JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

Preface

This special issue contains the selected papers chosen by guest editors from the proceedings of the *7th NATO RCMCIS 2005 Regional Conference on Military Communications and Information Systems*, Zegrze, Poland, 7–8 October 2005. The mission of the conference is to provide forum to disseminate and to discuss important topics for the military. These topics are currently investigated by the NATO research society and related to improvement of the military infrastructure for communications and information systems (CIS). The leading subject of the conference was *Technologies for the Military Transformation*. The majority of about 60 presented papers were related to the above mentioned main topic and they discussed the problems of the latest developments in systems and architectures, communication and information systems technologies, secure communications interoperability protocols, as well as cryptology.

Eight papers presented in this issue represent security and telecommunications technologies areas as discussed during the conference. For the security matters there are four papers. The first, by Christoph Karg and Martin Lies presents *A new approach to header compression in secure communications*. The authors propose a new header compression mechanism for the IPv6 protocol; the main benefit of the mechanism is a reduction of the overhead caused by IPSec in the tunnel mode which enlarges the datagrams in order to provide security services such as authentication and secrecy. The last two papers deal with the problems corresponding to design of the cipher systems. In the paper *Distrubution of the best nonzero differential and linear approximations of s-box functions* Krzysztof Chmiel discusses results of the effective designing of s-boxes for block ciphers; the author considers differential and linear approximations of two classes of s-box functions. Anna Grocholewska-Czuryło (*Random generation of Boolean functions with high degree of correlation immunity*) presents an algorithm that can generate randomly highly nonlinear resilient functions for the use mainly in stream cipher systems.

For the telecommunications technologies area there are five papers. The first paper is *End-to-end service survivability under attacks on networks* by Wojciech Molisz and Jacek Rak. The authors propose a model based on traffic parameters of a demand, like delay or bit rate, that allow establishing survivable attack-proof end-to-end connections. The next paper, *New model of identity checking in telecommunication digital channels* by Piotr Gajewski, Jerzy Łopatka and Zbigniew Piotrowski, describes a watermarking based technology system for correspondent identity verification in military telecommunication digital channels. *Planning the introduction of IPv6 in NATO* by Robert Goode is devoted to the areas which must be covered by the NATO IPv6 transition planning process in order to manage the introduction and migration to IPv6. In the next paper (*Simple admission control procedure for QoS packed switched military networks*) the authors, Damian Duda and Wojciech Burakowski, propose an admission control method based on the online traffic load measurements and take advantage from a possibility of the system over-provisioning. The final paper (*Performance evaluation of the multiple output queueing switch with different buffer arrangements strategy* by Grzegorz Danilewicz, Wojciech Kabaciński, Janusz Kleban, Damian Parniewicz, and Patryk Dąbrowski) deals with the problems of the switching system design.

Finally, we would like to take the opportunity to thank all authors of the presented papers as well as the referees who helped us with the evaluation process.

Wojciech Burakowski and Janusz Stokłosa
Guest Editors

Paper

A new approach to header compression in secure communications

Christoph Karg and Martin Lies

Abstract—The paper presents a new header compression mechanism for the IPv6 protocol. Its main benefit is the reduction of the overhead caused by IPSec tunnel mode which enlarges datagrams in order to provide security services such as authentication and secrecy.

Keywords—IPv6, header compression, IP security, restricted bandwidth.

1. Introduction

In contrast to civil networks, tactical military computer networks lack the broad data rate links. This is especially true for those tactical networks where the deployment is dependent on a high mobility. Because of the necessity to utilize wireless techniques or even open ISPs as transfer-nets, the topic of security becomes ever more important. The chosen security mechanism has to provide strong encryption of the transported data regardless of the deployed applications, has to support an authentication mechanism to limit the access and provide protection against a variety of attacks. These requirements are fulfilled by IPSec [2], as well as the need for a matching key exchange mechanism, the Internet key exchange (IKE) [6]. With these components, the communication of complete subnets can be secured using IPSec gateways, building virtual private networks (VPNs). In essence, every IP packet is completely encrypted inside another IP-packet, thereby hiding every information about the application, the transmitted data and also about the topology of the secured area.

The cost of the usage of IPSec tunnel mode is the growth in the required data rate for the same amount of transmitted data. To give an example, an Internet protocol Version 6 (IPv6) packet holds a header of 40 byte. IPSec tunnel mode adds 40 byte IP-header for the transfer in the black, i.e., unsecured network, then depending on the chosen authentication algorithm around 22 bytes for the authentication header (AH) and again depending on the chosen encryption algorithm another 22 bytes with a certain amount of padding to obtain the necessary length. This still ignores all application data.

Especially in narrow-bandwidth environments, e.g., communication across HF, GSM or satellite, the overhead caused by IPSec is not negligible. To enhance to performance of IPSec, a header compression mechanism was described in [5, 8]. The proposal resulted in the modification of an existing IPSec implementation. The release supported the following type of header compression. After a connection between two hosts inside two distinct secure subnets is established, the black header is replaced by a small index tag. Then the modified datagram is encrypted and encapsulated by the sending IPSec gateway. The receiving IPSec machine verifies the mandatory authentication. On success, it decrypts the payload and restores with the original red IPv6 header by replacing the index tag. Comprehensive tests substantiated the good performance of the header compression approach. Alas the approach has an major drawback. It is strictly link orientated and thus does not work in a network topology consisting of various (i.e., more than two) secure subnets.

This paper presents an enhancement of the above mechanism which works for arbitrary VPNs. It is developed to work within the network resource manager (NRM) [7]. An NRM is a security gateway with extended functionality. Besides enabling secure communications via IPSec, it gathers information on the current network traffic. This statistical data can be used to estimate the available bandwidth as well as the reliability of the existing connections. Furthermore, the NRM supplies an enhanced connection management in order to support secure multicast-based group communication and the corresponding key exchange mechanism (MIKE) as presented in [3].

The paper is organized as follows. In Section 2, the idea behind the header compression mechanism is described. The algorithms and data structures needed by header compression are presented in Sections 3 and 4, respectively. Section 5 provides information on how to integrate header compression into the IPSec framework. The performance of the compression mechanism is discussed in Section 6. The paper closes with concluding remarks in Section 7.

2. Header compression

The IP tunneling mechanism (see Fig. 1a for details) is an integral part of VPN solutions such as IPSec. A drawback is the enlarged datagram size, since two IP headers must be transmitted. In order to decrease the packet size, we propose to replace IP tunneling by an alternative header compression mechanism (see Fig. 1b). The idea is as follows. The sending gateway replaces the inner IP header by a compression header of smaller size. The compression header contains among other an unique identifier. This information enables the receiving gateway to reconstruct the original IP header and forward the datagram to its destination. The benefit comes from the fact that the identifier can be used often while the assignment between IP and compression header must be transmitted only a few times.

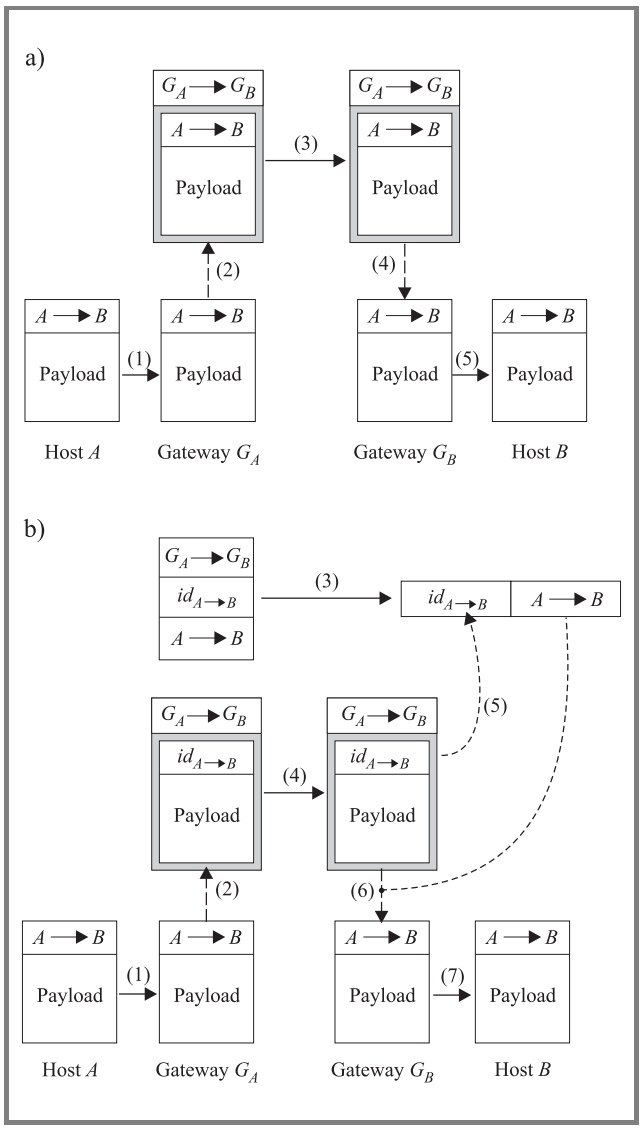


Fig. 1. (a) IP tunneling: (1) Host A sends an IP packet to host B via the gateways G_A and G_B, whereat the link between G_A and G_B is an IP tunnel. (2) On receipt, G_A stores the IP packet in a new one. This mechanism offers several opportunities to manipulate the IP packet to be tunneled. For example, IPSec encrypts the packet before transmission. (3) G_A sends the resulting packet to G_B. (4) G_B unfolds the incoming packet, this is, it removes the tunnel header and does some post-processing. For example, IPSec decrypts the payload and checks it't integrity. (5) Finally, G_B forwards the content to host B.
(b) Header compression: (1) Host A sends an IP packet to host B via the gateways G_A and G_B. Instead of tunneling the packet directly, gateway G_B performs the following manipulation. Based on the IP header $A \rightarrow B$, G_A computes an unique identifier $id_{A\rightarrow B}$. (2) G_A sends the pair $(A \rightarrow B, id_{A\rightarrow B})$ to gateway G_B, which stores the information in an internal lookup table. (3) G_A replaces the header $A \rightarrow B$ by the corresponding identifier $id_{A\rightarrow B}$ and sends the result to G_B. This replacement is done for all following IP packets with header $A \rightarrow B$. (4) On receipt, G_B removes the tunnel header and the identifier. (5) G_B uses $id_{A\rightarrow B}$ to reconstruct the original header $A \rightarrow B$. (6) G_B combines $A \rightarrow B$ with the payload. (7) G_B forwards the original IP packet to host B. If A sends another datagram to B, then the same identifier can be used. Hence, Step 3 can be omitted.

In the following we describe the header compression in more detail. There are several questions to be discussed.

- How do we compute the compression header and the identifier within?
- What is the structure of the header replacement?
- Which algorithms and data structures are required on sender and receiver side?
- How is header compression integrated in the security environment?

The following sections provide answers to these questions.

3. Computation of the compression header

The reduction of the datagram size is achieved by the replacement of the original IP header by a smaller compression header. Its structure depends on the algorithm to be used. There are two requirements to be met. Firstly, the information provided by compression header must be unique such that the receiver can reconstruct the original header. Secondly, the identifier should be usable for many different IP headers so that the amount of assignment messages is minimized.
A standard way to derive an unique identifier is universal hashing with open addressing (see [4] for an excellent discussion of this topic). If the range of the hash function fits into a 4-byte integer, there is a possibility of 2^{32} simultaneous outgoing transmissions. If the hash function is chosen uniformly at random then a good average case performance is guaranteed.
To determine the input of the hash function, let's take a look on the structure of the IPv6 header depicted in Fig. 2. The header size is 40 bytes. The largest part consists of the source and destination address each with a size of 16 bytes. Consider for a moment only datagrams sent from the host A to the host B. In this case, the source and destination address are constant. The contents of the fields payload length, next header, and hop limit may change frequently. The frequency of change of the contents stored in the fields traffic class and flow label is difficult to estimate. It depends on the quality of service features of the underlying network. As a rule of thumb, we assume a low modification rate.
Based on the above assumption, the fields version, flow label, traffic class, source address and destination address are the input of the hash function. With other words, the hash function maps 32 bytes to values of 4 byte length. Open addressing guarantees unique hash keys given the number of different inputs is smaller than the range (i.e., 2^{32}). Since the payload length, hop limit and next header entries do not influence the outcome of the hash function, they additionally must be stored in the compression

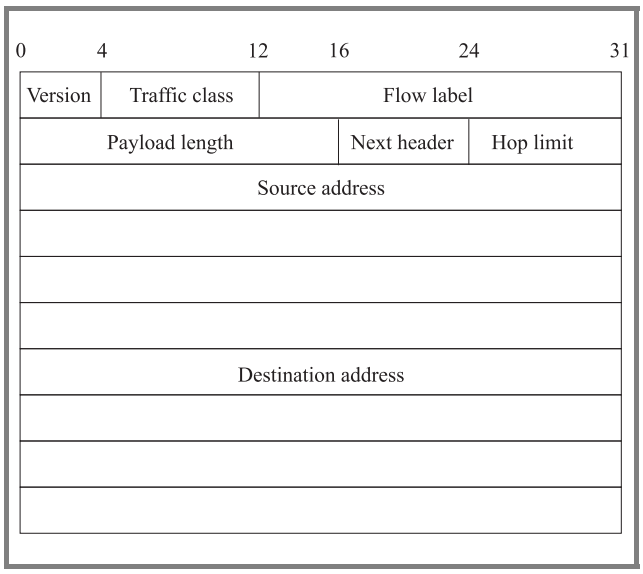


Fig. 2. Format of the IPv6 header.

header. This leads to the following structure of the compression header:

1) hash key to determine version, flow label, traffic class, source address and destination address;
2) payload length;
3) next header;
4) hop limit.

The size of the compression header is 8 bytes. Hence, it's size is 20% of the IP header.
Additional costs arise from the fact that the assignment between hash key and according IP header parts must be transmitted to the receiving gateways. The cost in terms of data rate depends on the way, the assignment information is distributed. Among others, two following methods are conceivable.

- If the compression starts, the first datagrams are not compressed. Instead, they are extended with an optional header which contains the hash key. On receipt, the destination gateway stores this identifier together with the associated parts of the IP header. On this way, it learns how to decompress future datagrams.
- In the beginning of compression, the first datagrams are transmitted without modification. Additionally, the assignment information is sent in a separate datagram.

Note, the transmission of the compression information is critical for the whole communication. Hence, the spread of the assignment information must be redundant to guarantee that the destination gateway gets the assignment before the compressed data.

The computation of the index is strictly sender oriented. This is, the sending gateway computes the identifiers of all outgoing datagrams independently on it's own. As an advantage, the gateways do not need to coordinate the identifier creation. This keeps things simple and there is no additional communication between the gateways required. Furthermore, compression header assignment messages are receivable even by those gateways which are in emission control (emcon) condition.

4. Data structures

The mapping between identifiers and according IP header fragments must be stored in an appropriate data structure. The requirement to be met is the fast lookup of a given index. The best performance is achieved by an hash table. However this is not feasible because of the large range of the deployed hash function and its many memory consumption. A good compromise between space and performance is the usage of a balanced search tree such as red-black trees [4]. Such a data structure guarantees time $0(\log_2 n)$ for search, insertion and deletion where n is the number of elements stored in the tree.
In order to compress a datagram the sending gateway acts as follows. It computes the hash index of the IP header fragment. Then, it replaces the header by the compression header. If it is the first time, that this header fragment is compressed, then the assignment information is sent the receiving gateway. Additionally, the pair consisting of index and header fragment is stored in the search tree.
On the receiving side, the data keeping is slightly different. Since the identifier is correlated with the gateway which created it, the receiver must manage a search tree for each sending gateway. These trees are stored in another search tree with the senders' IP addresses as key.
The contents of the search trees change dynamically. An entry may be deleted if the respective hash index was not used for a certain time. This behavior increases the performance of both running time and memory usage.

5. Integration into security services

The header compression mechanism is thought to be used in combination with a secure transmission method. An obvious approach is the IP security protocol suite [2]. Particularly, the compressed datagrams are sent via IPSec transport mode. The usage of IPSec has the advantage of well-developed administrative tools such as automated key negotiation. However, the header compression must be integrated in the IPSec processing chain. There are two possible solutions. The first one is the extension of the operating system's kernel. The second one consists in the packet filtering and manipulation in user space via the TUN/TAP interface.
The integration of header compression into the operating system results in optimal performance since the code runs

in kernel space. However for modification of the kernel the underlying source code is required. Even if the code is available, the compression algorithm must be adapted to the respective conditions of the operating system. Hence, the work is not portable to other operating systems. Another drawback is the large amount of maintenance required to keep the software up to date with new kernel releases.

The second approach uses the TUN/TAP interface for packet manipulation. TUN/TAP is a standardized mechanism for manipulation of IP packets and Ethernet frames. It is supported by major UNIX systems such as BSD, Linux and Solaris and even by Windows XP. Hence, the code of the header compression extension is portable between the different operating systems with moderate effort. A potential disadvantage is a loss in performance since the compression runs in user space and has to share the system's resources with the other processes.

Finally, we remark that IPSec is not the only way to deploy header compression. An alternative is OpenVPN [1], a VPN solution which uses the SSL/TLS protocol for a secure data tunneling. Compared to IPSec, OpenVPN is a lightweight VPN solution. It can be set up easily. Additionally, OpenVPN provides a traffic shaping functionality to restrict the rate of data flowing to the tunnel.

6. Performance estimates

To evaluate the performance of the header compression mechanism, two aspects have to be considered:

- **Network delay**. The compression of the IP header is an additional processing step in the data flow which increases the delay between sending and receiving host.
- **Data rate improvement**. Header compression is deployed with the goal of reducing the amount of data to be transmitted. Hence, this criterion has to analyzed.

The influence on the network delay can be estimated by the amount of time needed to compute the compression header. The expensive part is the lookup of the hash index in the balanced search tree.

A well-known result concerning universal hashing is the following [4]. If the hash function h is randomly chosen from a universal collection of hash functions and is used to hash n keys into a table of size m, where $m \leq n$, the expected number of collisions involving a particular key is less than 1. Hence, if the outgoing connections result in n different hash indices, where $n \leq 2^{32}$, then there are no collisions in the average case. Since the keys are stored in a balanced tree, the expected running time per lookup operation is logarithmic in n.

What does this mean in practice? Assume, that the search tree is based on red-black trees. Then a tree storing n hash indices as height at most $2\log_2(n+1)$. This value is an upper bound for the cost of a lookup operation. Each entry consists of at least 40 bytes (4 bytes for the index and 36 for the IP header fragment). This is a lower bound for the memory needed to store the search tree. If any host in the subnet has 10000 concurrent outgoing data connections then the number of different hash indices is at most 10000. Given the total number n of hosts, the upper bound of different indices is $10000n$. Using this bound, we estimate the size of the tree for concrete values for n (Table 1).

Table 1
Tree size estimations

Hosts	Indices	Tree height	Size [MB]
10	100000	34	3.815
20	200000	36	7.629
50	500000	38	19.073
100	1000000	40	38.147
200	2000000	42	76.294
500	5000000	45	190.735

Because the number of steps to compute a table lookup is small even for a large number of hosts, the delay caused by header compression is negligible on modern computer hardware.

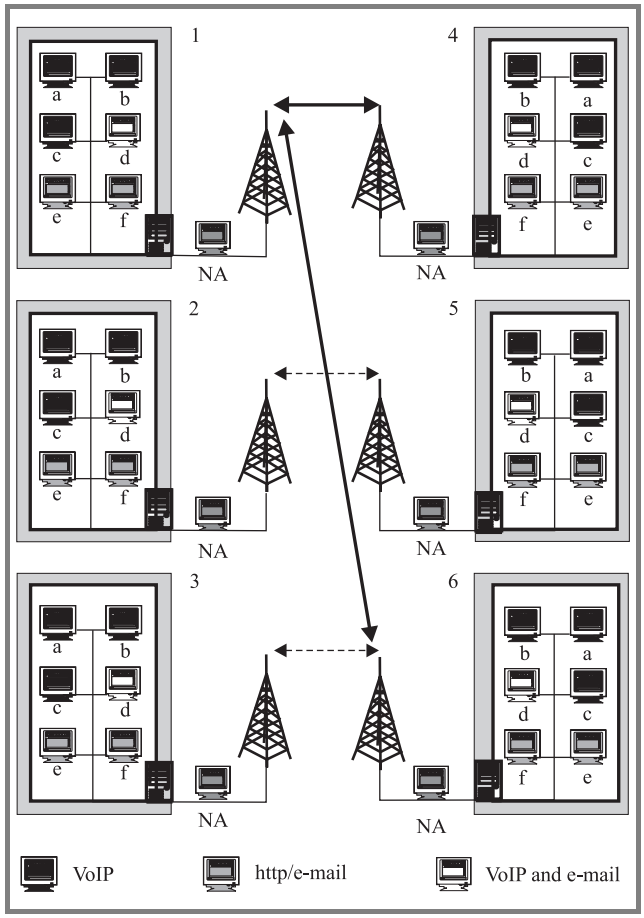


Fig. 3. The topology of a tactical military network at batallion level.

The improvement of the data rate is quite difficult to estimate. It depends on both the topology of the underlying network and the type of traffic and its distribution within the network. The topology can be created based on existing military tactical networks. For an example, we refer to Fig. 3. The determination of network traffic structure in terms of distribution and service type is a difficult or even impossible venture.
A promising approach consists in a statistical analysis via a network simulation which is beyond the scope of this paper. As an obvious rule of thumb, the gain of header compression is high for network services which send datagrams in an high frequency. A good example is voice over IP (VoIP).

7. Conclusions and future work

This paper presents a new approach to header compression to be used in combination with the IP security framework. The benefit is a reduction of the overhead caused by IPSec tunnel mode in terms of enlarged datagrams. Furthermore, it can be integrated in common existing operating systems with moderate effort. Currently the header compression is a concept. Future steps are simulations based on network simulator 2 and the implementation in a testing environment.

References

[1] OpenVPN. Project webpage, http://www.openvpn.net

[2] R. Atkinson and S. Kent, "Security architecture for the Internet protocol", RFC 2401, Nov. 1998.

[3] T. Aurisch and C. Karg, "A daemon for multicast Internet key exchange", in *IEEE Conf. Loc. Comput. Netw.*, Bonn, Germany, 2003, pp. 368–376.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge: MIT Press, 2003.

[5] R. Göbel, "Beschreibung eines QoS-unterstützten Netzadapters für schmalbandige Subnetztypen", FKIE-Bericht 38, FGAN, Jan. 2002.

[6] D. Harkins and C. Carrel, "The Internet key exchange (IKE)", RFC 2409, Nov. 1998.

[7] M. Lies, P. Sevenich, C. Karg, and C. Barz, "Resource management in tactical military networks", in *NATO/RTO IST Symp. Milit. Commun.*, Roma, Italy, 2005.

[8] P. Sevenich and G. Beling, "Multiplexing time-critical data over heterogeneous subnetworks of low bandwidth", in *Reg. Conf. Milit. Commun. Inform. Syst. RCMCIS*, Zegrze, Poland, 1999.

Christoph Karg is a Professor of computer science at the University of Applied Sciences at Aalen, Germany. Doctor's Karg research interests are computer networks, cryptology, algorithms and complexity theory.

e-mail: christoph.karg@htw-aalen.de
Fakultät Elektronik und Informatik
HTW Aalen
Beethovenstraße 1
73430 Aalen, Germany

Martin Lies is a senior scientist of the communications department of FGAN/FKIE. His research interests are computer networks with special emphasis on security and resource restrictions, robotics and cryptology.

e-mail: lies@fgan.de
Department Computer Networks
FGAN
Neuenahrer Straße 20
53343 Bonn, Germany

Paper

Distribution of the best nonzero differential and linear approximations of s-box functions

Krzysztof Chmiel

Abstract— In the paper the differential and the linear approximations of two classes of s-box functions are considered. The classes are the permutations and arbitrary functions with n binary inputs and m binary outputs, where $1 \le n = m \le 10$. For randomly chosen functions from each of the classes, the two-dimensional distributions of the best nonzero approximations are investigated. The obtained results indicate that starting from some value of n, the linear approximation of s-box functions becomes more effective than the differential approximation. This advantage of the linear approximation rises with the increase of n and for DES size s-boxes is not yet visible.

Keywords— differential cryptanalysis, linear cryptanalysis, substitution boxes.

1. Introduction

Differential and linear cryptanalysis belong to main topics in cryptology since they were introduced and successfully applied to the data encryption standard (DES). Unlike the differential cryptanalysis, which is essentially a chosen-plaintext attack [1, 10, 11], the linear cryptanalysis is essentially a known-plaintext attack and moreover is applicable to an only-ciphertext attack under some circumstances [2–12].

The basic idea of differential cryptanalysis is to analyze the effect of particular differences in plaintext pairs on the differences of the resultant ciphertext pairs. The differences are usually calculated as a result of XOR operation. Input XOR of a cipher algorithm causes a specified output XOR with some probability. The appropriate, approximate expression will be called the differential approximation. By the *differential approximation* of function $Y = f(X) : \{0,1\}^n \to \{0,1\}^m$ we mean an arbitrary equation of the form:

$$f(X) \oplus f(X \oplus X') = Y',$$

which is fulfilled with approximation probability $p = N(X',Y')/2^n$, where $X' \in \{0,\ldots,2^n-1\}, Y' \in \{0,\ldots,2^m-1\}$ and $N(X', Y')$ denotes the number of input pairs $(X, X \oplus X')$ for which the equation holds. The numbers X', Y' are called input and output *difference* respectively and the function $N(X',Y')$ is called the *counting function* of the approximation. The magnitude of p represents the *effectiveness* of the approximation. Among approximations we distinguish the *zero differential approximation* with $X' = Y' = 0$, which probability p is equal to 1 for arbitrary function f.

The basic idea of linear cryptanalysis is to describe a given cipher algorithm by a linear approximate expression, so-called linear approximation. In general, the *linear approximation* of function $Y = f(X) : \{0,1\}^n \to \{0,1\}^m$ is defined as an arbitrary equation of the form:

$$\bigoplus_{i \in Y'} y_i = \bigoplus_{j \in X'} x_j,$$

which is fulfilled with approximation probability $p = N(X',Y')/2^n$, where $X' \subseteq \{1,\ldots,n\}$, $Y' \subseteq \{1,\ldots,m\}$ and $N(X',Y')$ denotes the number of pairs (X,Y) for which the equation holds. The sets of indexes X',Y' are called input and output *mask* respectively and the function $N(X',Y')$ is called the *counting function* of the approximation. The *effectiveness* of the approximation is represented by magnitude of $|\Delta p| = |p - 1/2|$. By the *zero linear approximation* we mean approximation with $X' = Y' = \Phi$, which probability p is equal to 1 for arbitrary function f. Masks X',Y' are often denoted by numbers, corresponding to the zero-one representation of sets.

The set of all differential approximations of function f can be described in the form of the *approximation table* TDf, called in [1] the *difference distribution table*. The element $TDf[X',Y']$ of the table, is defined as follows:

$$TDf[X',Y'] = N(X',Y').$$

The maximum value of TDf, that corresponds to the best, i.e., most effective, nonzero differential approximation, is denoted by $\max TD$ and is defined by formula:

$$\max TD = \max\{TDf[X',Y'] : X' \neq 0 \vee Y' \neq 0\}.$$

Similarly, the set of all linear approximations of function f is represented in the form of the *approximation table* TAf. The element $TAf[X',Y']$ of the table, is defined as follows:

$$TAf[X',Y'] = \Delta N(X',Y') = N(X',Y') - 2^{n-1}.$$

The maximum absolute value of TAf, which corresponds to the best nonzero linear approximation, is denoted by $\max TA$ and is defined in the following way:

$$\max TA = \max\{|TAf[X',Y']| : X' \neq \Phi \vee Y' \neq \Phi\}.$$

The approximation tables of an example function f are presented in Table 1. There exist many effective approximations of the function, identified by nonzero values of

Table 1
Function f and its approximation tables TDf and TAf ($n = 4$, $m = 2$)

f		TDf					TAf				
X	$Y = f(X)$	X'	Y'				X'	Y'			
			0	1	2	3		0	1	2	3
0	3	0	16	0	0	0	0	8	–2	–1	1
1	3	1	10	0	2	4	1	0	–2	1	–1
2	3	2	6	0	2	8	2	0	0	1	1
3	0	3	6	0	2	8	3	0	0	3	–1
4	1	4	2	8	6	0	4	0	0	–1	**7**
5	3	5	2	8	6	0	5	0	0	–3	1
6	1	6	0	2	**12**	2	6	0	2	1	–1
7	1	7	2	4	10	0	7	0	2	–1	1
8	0	8	4	2	0	10	8	0	–4	1	1
9	0	9	2	0	2	**12**	9	0	0	–1	–1
10	3	10	8	2	0	6	10	0	–2	–5	1
11	3	11	8	2	0	6	11	0	2	1	–1
12	1	12	0	6	8	2	12	0	2	–3	–1
13	2	13	0	6	8	2	13	0	–2	–1	1
14	2	14	2	8	6	0	14	0	–4	–1	–1
15	2	15	2	**12**	2	0	15	0	0	1	1

the tables. The best nonzero differential approximations have $\max TD = 12$ and probability $p = 12/16$, while the best nonzero linear approximation has $\max TA = 7$ and probability $|\Delta p| = 7/16$.

The size of the approximation tables TDf and TAf of function f is equal to 2^{n+m} and the basic algorithms compute a single element of the tables in exponential time. The used in the investigation fast algorithms, presented in detail in [10], compute the approximation tables in time at worst linear for a single element, without memory needed for storage of the whole table.

2. Results

The presented in this chapter results of experiments concern the distribution of the best nonzero differential and linear approximations of two classes of s-box functions $Y = f(X)$. The classes are the permutations and arbitrary functions of the type $f : \{0,1\}^n \to \{0,1\}^m$, for $1 \le n = m \le 10$. For each value of n, the investigation was carried out for 1000 randomly chosen functions from the class. For each function, with use of the mentioned in the previous chapter fast algorithms, were calculated values of $\max TD$ and $\max TA$. Distribution of pairs $(\max TD, \max TA)$ was the goal of the computation. The obtained results are presented in Figs. 1–19.

For $n = m = 1$ (Fig. 1), the proportional distributions obtained for permutations and arbitrary functions are identical. For 100% of functions, from each of the classes, the obtained pair $(\max TD, \max TA)$ is equal to (2, 1).

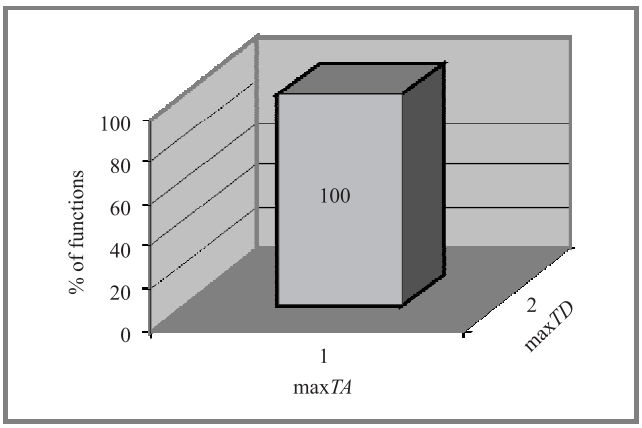


Fig. 1. Distribution for permutations and arbitrary functions ($n = 1$, $m = 1$).

For $n = m = 2$ (Figs. 2 and 3), the distributions for permutations and arbitrary functions differ. For 100% of permutations, the obtained pair $(\max TD, \max TA)$ is equal

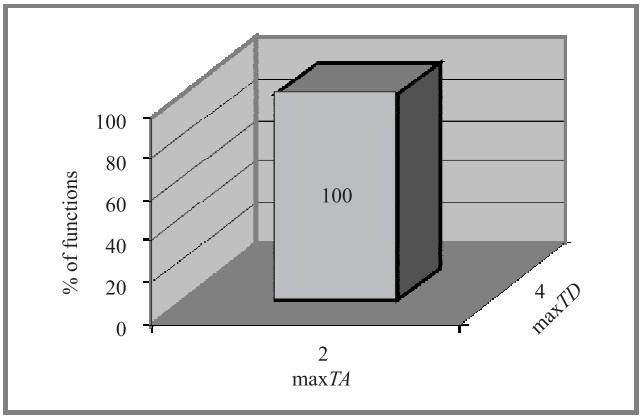


Fig. 2. Proportional distribution for permutations ($n = 2$, $m = 2$).

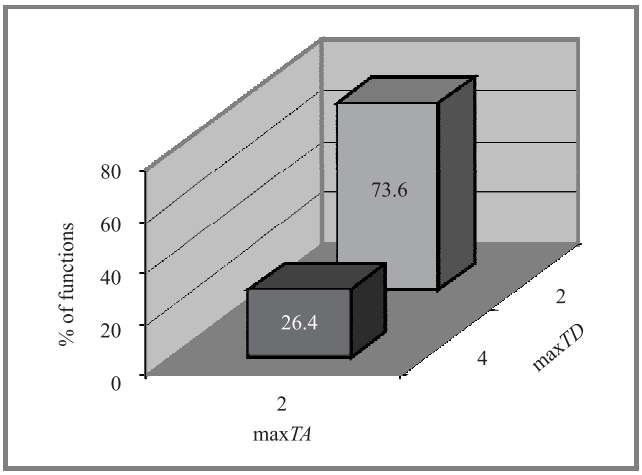


Fig. 3. Proportional distribution for arbitrary functions ($n = 2$, $m = 2$).

to (4, 2). For arbitrary functions, the same pair (4, 2) is obtained for 26.4% of functions while for remaining 73.6% of functions is obtained pair (2, 2). The results indicate, that

resistance to linear approximation of permutations and arbitrary functions with two input and output bits is the same, while about 3/4 of arbitrary functions are more resistant to differential approximation than permutations.

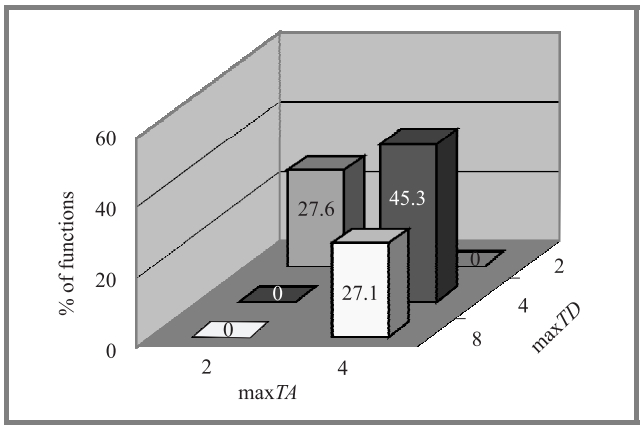


Fig. 4. Proportional distribution for permutations ($n = 3$, $m = 3$).

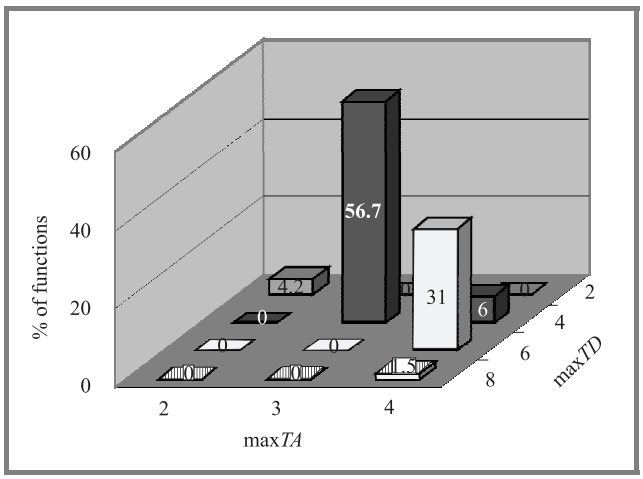


Fig. 5. Proportional distribution for arbitrary functions ($n = 3$, $m = 3$).

For $n = m = 3$ (Figs. 4 and 5), three different pairs $(\max TD, \max TA)$ are obtained for permutations while for arbitrary functions are obtained five different pairs, among which two pairs are dominant. Among permutations there are more functions with pair (8, 4) that are easiest to approximate as well as more functions with pair (2, 2) that are most difficult to approximate, than among arbitrary functions. It should be noticed, that for permutations the values of $\max TA$ are even while for arbitrary functions are odd as well. The values of $\max TD$ are even both for permutations and arbitrary functions.

For $n = m = 4$ (Figs. 6 and 7), there exist for permutations two dominant pairs and for arbitrary functions also two, but not the same. Both distributions have the evident maximum, which is obtained for the pair $(\max TD, \max TA)$ equal to (6, 6).

For $n = m = 5$ (Figs. 8 and 9), there are visible bars in the diagrams for the values of $\max TD$ equal to 6, 8 and 10.

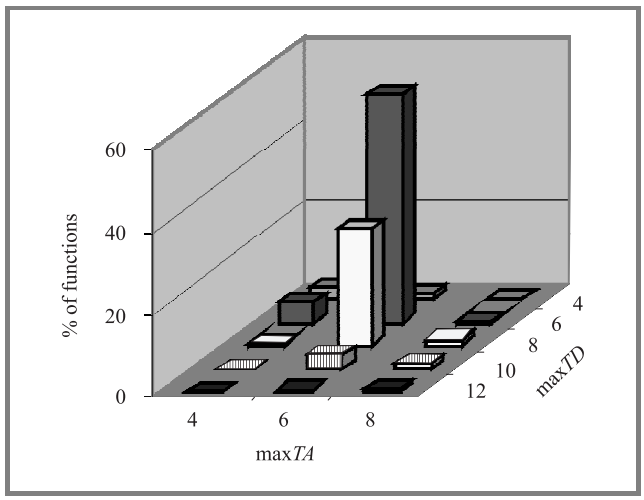


Fig. 6. Proportional distribution for permutations ($n = 4$, $m = 4$).

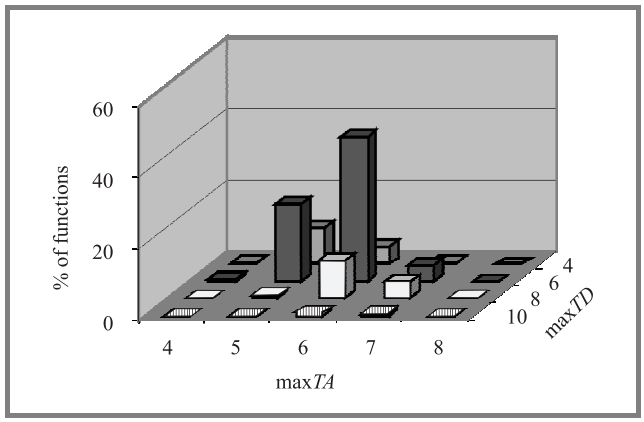


Fig. 7. Proportional distribution for arbitrary functions ($n = 4$, $m = 4$).

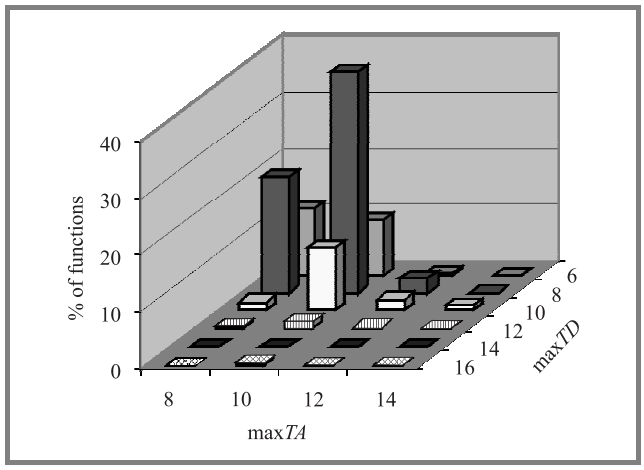


Fig. 8. Proportional distribution for permutations ($n = 5$, $m = 5$).

The maximum of distribution is less for arbitrary functions, because of the even and odd values of $\max TA$.

For $n = m = 6$ (Figs. 10 and 11), there are visible in the distributions for permutations and arbitrary functions, two significant series of results for $\max TD$ equal to 8 and 10.

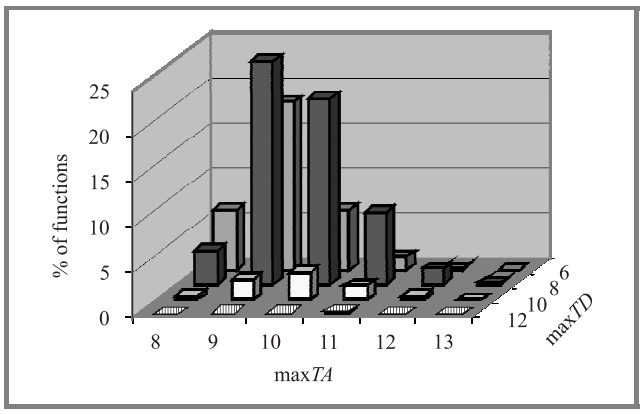


Fig. 9. Proportional distribution for arbitrary functions ($n = 5$, $m = 5$).

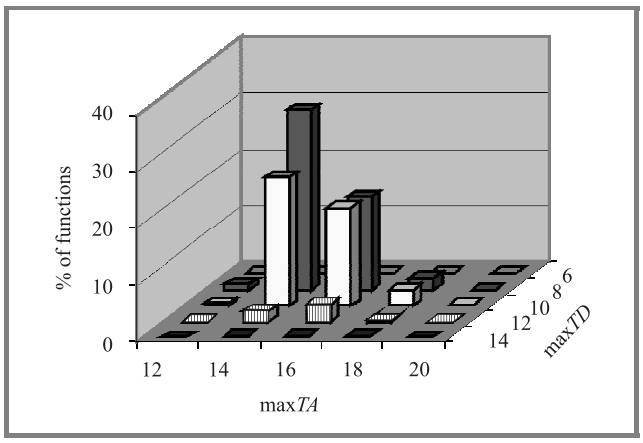


Fig. 10. Proportional distribution for permutations ($n = 6$, $m = 6$).

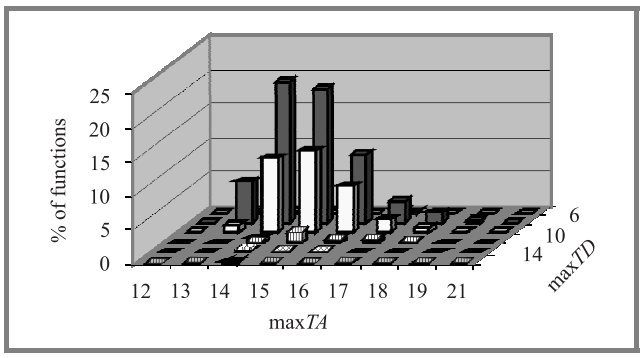


Fig. 11. Proportional distribution for arbitrary functions ($n = 6$, $m = 6$).

For $n = m = 7$ (Figs. 12 and 13), there are visible three series of results for $\max TD$ equal to 8, 10 and 12, in the distributions for permutations and arbitrary functions. The middle series is clearly dominant.

For $n = m = 8$ (Figs. 14 and 15), there are visible in the distributions for permutations and arbitrary functions, two significant series of results for $\max TD$ equal to 10 and 12. The series are rather equivalent this time. No one of them dominates.

For $n = m = 9$ (Figs. 16 and 17), in the distributions for permutations and arbitrary functions, are visible two series

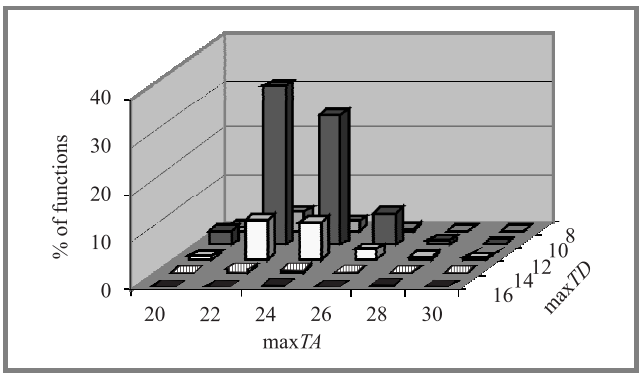


Fig. 12. Proportional distribution for permutations ($n = 7$, $m = 7$).

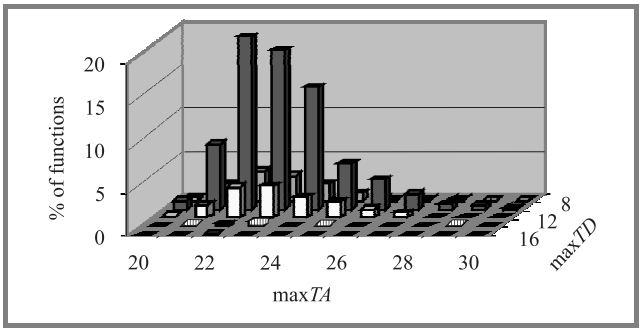


Fig. 13. Proportional distribution for arbitrary functions ($n = 7$, $m = 7$).

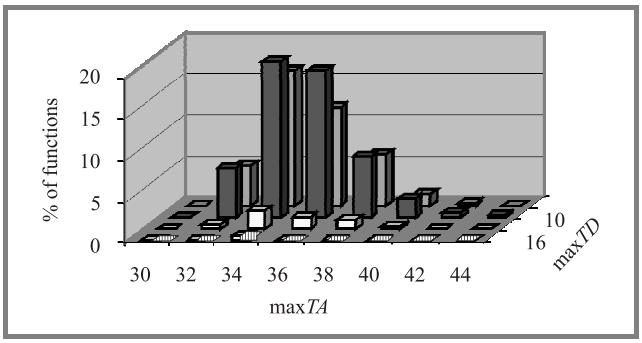


Fig. 14. Proportional distribution for permutations ($n = 8$, $m = 8$).

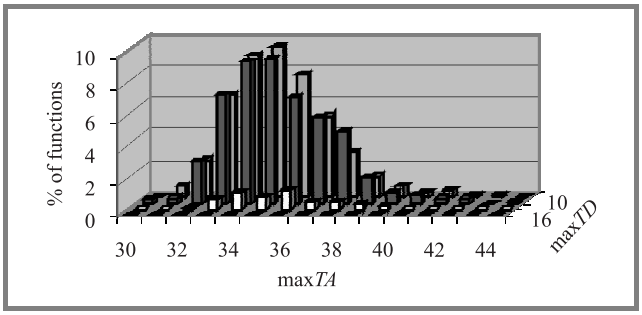


Fig. 15. Proportional distribution for arbitrary functions ($n = 8$, $m = 8$).

of results for $\max TD$ equal to 12 and 14. The series for $\max TD$ equal to 12 is clearly dominant.

For $n = m = 10$ (Figs. 18 and 19), there are visible two significant series of results for $\max TD$ equal to 12 and 14,

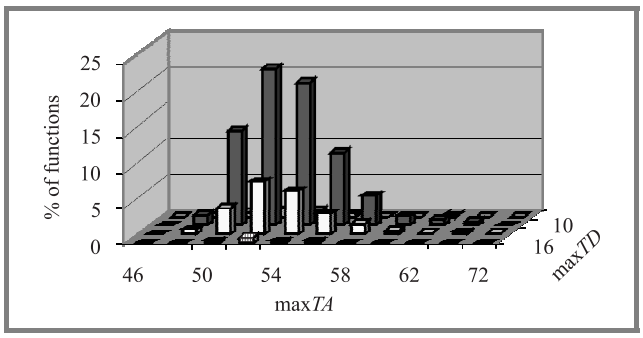


Fig. 16. Proportional distribution for permutations ($n = 9$, $m = 9$).

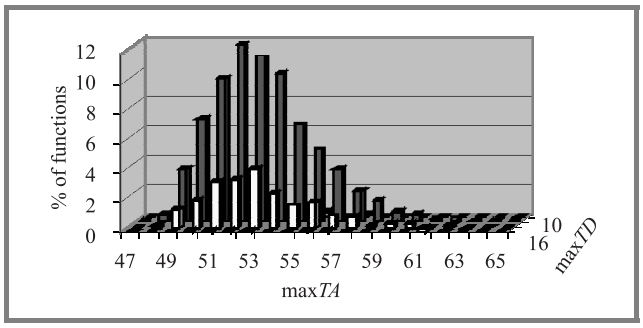


Fig. 17. Proportional distribution for arbitrary functions ($n = 9$, $m = 9$).

in the distributions for permutations and arbitrary functions. The series for value 14 of $\max TD$ is not so dominant like in the case of $n = m = 9$.

Considering the results for $1 \leq n = m \leq 10$, presented in Figs. 1–19 we can observe, that the significant for distributions ranges of $\max TD$ and $\max TA$ as well as the values of pairs $(\max TD, \max TA)$ for which are obtained maxima

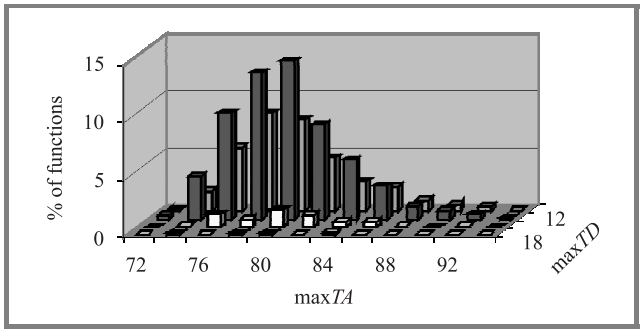


Fig. 18. Proportional distribution for permutations ($n = 10$, $m = 10$).

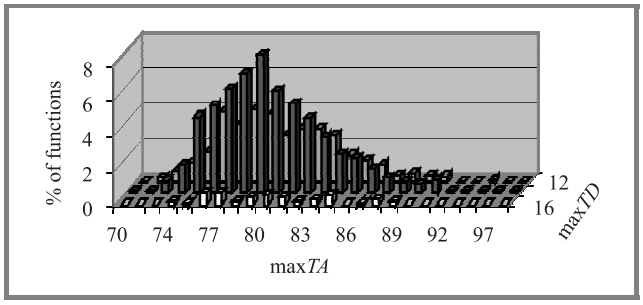


Fig. 19. Proportional distribution for arbitrary functions ($n = 10$, $m = 10$).

of distributions, are about the same for permutations and arbitrary functions. The values of maxima are greater for permutations. It follows from the fact, that for $n \geq 3$, the values of $\max TA$ are even for permutations while for arbitrary functions are odd as well. Thus, we can say that the results obtained for permutations and arbitrary functions are similar.

Comparing the differential and the linear approximation we can observe, that the ranges of $\max TD$ are narrow while the ranges of $\max TA$ are wide. With the increase of $n = m$ the values of $\max TA$ rise much faster than the values of $\max TD$. It means that the linear approximation of s-box functions becomes much more effective than the differential approximation. This advantage of the linear approximation starts at some value of $n = m$ and rises with the increase of this value.

3. Results for DES size s-boxes

The presented in this chapter results concern the distribution of the best nonzero differential and linear approximations of permutations with 6 input bits and 4 output bits. Similarly to definition of DES s-boxes, by permutation in this case we mean a set of four 4-bit permutations. In general, for $n > m$, by permutation we mean in fact a set of 2^{n-m} m-bit permutations. The results in detail are presented in Table 2 and illustrated in Fig. 20.

Table 2

Results of experiments for permutations – DES size ($n = 6$, $m = 4$)

$\max TD$	$\max TA$					Total
	10	12	14	16	18	
12	0	6	5	3	0	14
14	1	144	141	33	4	323
16	1	107	255	65	12	440
18	0	24	94	41	5	164
20	0	3	28	14	2	47
22	0	3	3	4	1	11
24	0	0	0	0	1	1
Total	2	287	526	160	25	1000

For DES size s-boxes, the advantage of the linear approximation over the differential one is not yet visible. The range of $\max TD$ is from 12 to 24 and the range of $\max TA$ is from 10 to 18. So the values of $\max TD$ and $\max TA$ are comparable.

The distribution of the best nonzero approximations enables to evaluate the quality of constructed s-boxes. The less the values of $\max TD$ and $\max TA$ the better is the s-box. The quality of DES s-boxes *S1–S8* is presented in Table 3. The value of $\max TD$ for all s-boxes is equal to 16. The best s-box of DES is *S6* with $\max TA = 14$ and the worst is *S5* with $\max TA = 20$.

It follows from Table 2, that for 25.5% of randomly selected s-boxes, the obtained pair $(\max TD, \max TA)$ is equal to (16, 14). Thus, parameters of s-box *S6* correspond to

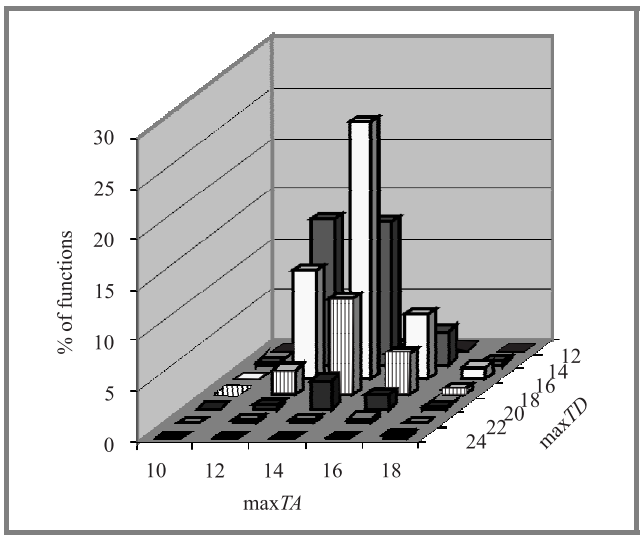


Fig. 20. Proportional distribution for permutations – DES size ($n = 6$, $m = 4$).

Table 3
Quality of DES s-boxes *S1-S8*

max TD	max TA					
	10	12	14	16	18	20
12						
14						
16			*S6*	*S2* *S3* *S4* *S8*	*S1* *S7*	*S5*

the maximum of the distribution. There are 40.5% of s-boxes with parameters better than of s-box *S6*. We have obtained, that among three randomly selected s-boxes, two of them are not worse than the best s-box of DES *S6* and one of them is better. On the other hand, the value 20 of max TA of the worst s-box *S5*, was not obtained for any of the 1000 randomly selected s-boxes. The quality of DES s-boxes is obviously not the best possible one.

4. Conclusion

The basic algorithms to compute a single element of the approximation tables TDf and TAf are of exponential complexity. The presented in [10] fast algorithms compute the values of max TD and max TA in at worst linear time for a single element, without memory needed for storage of the whole table. The fast algorithms were used to calculate the distribution of pairs $(\max TD, \max TA)$ for randomly chosen permutations and arbitrary functions with n binary inputs and m binary outputs, where $1 \leq n = m \leq 10$. For both classes of functions, the obtained results were similar. The main conclusion is that starting from some value of n, linear approximation of s-box functions becomes much more effective than differential approximation. Moreover, this advantage of linear approximation rises with the increase of n and for DES size s-boxes is not yet visible.

References

[1] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*. New York: Springer-Verlag, 1993.

[2] K. Chmiel, "Linear cryptanalysis of the reduced DES algorithms", in *Proc. Reg. Conf. Milit. Commun. Inform. Syst. 2000*, Zegrze, Poland, 2000, vol. 1, pp. 111–118.

[3] K. Chmiel, "Linear approximation of some s-box functions", in *Proc. Reg. Conf. Milit. Commun. Inform. Syst. 2001*, Zegrze, Poland, 2001, vol. 1, pp. 211–218.

[4] K. Chmiel, "On some models of arithmetic sum function linear approximation", in *Proc. NATO Reg. Conf. Milit. Commun. Inform. Syst. 2002*, Zegrze, Poland, 2002, vol. 2, pp. 199–204.

[5] K. Chmiel, "Linear approximation of arithmetic sum function", in *Artificial Intelligence and Security in Computing Systems*, J. Sołdek and L. Drobiazgiewicz, Eds. Boston: Kluwer, 2003, pp. 293–302.

[6] K. Chmiel, "Linear approximation of arithmetic subtraction function", in *Proc. NATO Reg. Conf. Milit. Commun. Inform. Syst. 2003*, Zegrze, Poland, 2003, pp. 1–6.

[7] K. Chmiel, "Linear approximation of structures with selectors", in *Proc. 6th NATO Reg. Conf. Milit. Commun. Inform. Syst. 2004*, Zegrze, Poland, 2004, pp. 269–273.

[8] K. Chmiel, "On arithmetic subtraction linear approximation", in *Enhanced Methods in Computer Security, Biometric and Artificial Intelligence Systems*, J. Pejaś and A. Piegat, Eds. New York: Kluwer, 2005, pp. 125–134.

[9] K. Chmiel, "Fast computation of approximation tables", in *Information Processing and Security Systems*, K. Saeed and J. Pejaś, Eds. New York: Springer, 2005, pp. 125–134.

[10] K. Chmiel, "Differential and linear approximation of s-box functions", in *Image Analysis, Computer Graphics, Security Systems and Artificial Intelligence Applications*, K. Saeed *et al*., Eds. Białystok: University of Finance and Management in Białystok, 2005, vol. 1, pp. 191–200.

[11] A. Górska, K. Górski, Z. Kotulski, A. Paszkiewicz, and J. Szczepański, "New experimental results in differential – linear cryptanalysis of reduced variants of DES", in *Proc. 8th Int. Conf. Adv. Comput. Syst. ACS'2001*, Mielno, Poland, 2001, vol. 1, pp. 333–346.

[12] M. Matsui, "Linear cryptanalysis method for DES cipher", in *Advances in Cryptology Eurocrypt'93*, T. Helleseth, Ed. New York: Springer-Verlag, 1994, pp. 386–397.

Krzysztof Chmiel is an adjunct at Poznań University of Technology, Poland. His research and scientific interests focus on data security in information systems and cryptology, especially methods of designing and cryptanalysis of cryptographic algorithms. He is author of a number of publications on differential and linear approximation of block ciphers and their component functions.

e-mail: Chmiel@sk-kari.put.poznan.pl
Institute of Control and Information Engineering
Poznań University of Technology
Marii Skłodowskiej-Curie Sq. 5
60-965 Poznań, Poland

Paper

Random generation of Boolean functions with high degree of correlation immunity

Anna Grocholewska-Czuryło

Abstract—In recent years a cryptographic community is paying a lot of attention to the constructions of so called resilient functions for use mainly in stream cipher systems. Very little work however has been devoted to random generation of such functions. This paper tries to fill that gap and presents an algorithm that can generate at random highly nonlinear resilient functions. Generated functions are analyzed and compared to the results obtained from the best know constructions and some upper bounds on nonlinearity and resiliency. It is shown that randomly generated functions achieve in most cases results equal to the best known designs, while in other cases fall just behind such constructs. It is argued that the algorithm can perhaps be used to prove the existence of some resilient functions for which no mathematical prove has been given so far.

Keywords—cryptography, ciphers, Boolean functions, correlation immunity, reslilience, random generation.

1. Introduction

Boolean functions play an important role in virtually any modern cryptographic system – be it block or stream ciphers, private or public key systems, authentication algorithms, etc. As security of these systems relies on Boolean functions these functions should posses some specific criteria that would protect a cryptographic system from any existing cryptanalytic attacks, and preferably make it also immune against any attacks that might be designed in the future. These criteria are called cryptographic criteria.

It is widely accepted among cryptologists that most important criteria are balancedness, high nonlinearity, propagation criteria, correlation immunity, high algebraic degree. Unfortunately no Boolean function exists that would fulfil all of these criteria to the maximum, so finding a cryptographically strong Boolean functions is always a trade-off between these criteria and is not a trivial task.

In particular, a functions whose output leaks no information about its input values is of great importance. Such functions are called correlation immune Boolean functions and were introduced by T. Siegenthaler in 1984 [32] and ever since then have been a topic of active research. A balanced correlation immune function is called a resilient function. As balancedness is one criterion that should be fulfilled under any circumstances, resilience is a criterion most often mentioned in the scientific literature when one talks about correlation immunity.

Most of the cryptographic criteria is in one way or another related to nonlinearity of the Boolean function. Highest nonlinearity is very desirable so most of the research concentrates on fulfilling the cryptographic criteria while maintaining a highest possible nonlinearity, which very often (virtually always) has to be sacrificed to some extent.

The approach to finding a good cryptographic functions is most often based on specific algebraic constructions of Boolean functions with desirable properties – like highly nonlinear Boolean function with high order of resiliency. Or constructing bent functions (functions with highest possible nonlinearity) and then modifying them to fulfil other cryptographic criteria.

In the article the author argues that the use of randomly chosen Boolean functions with good cryptographic properties (if we are able to find such functions) is probably better than the use of functions with similar parameters which are obtained by explicit constructions. The main reason is that explicit constructions usually lead to functions which have very particular (algebraic or combinatorial) structures, which may induce weaknesses regarding existing or future attacks. Therefore, author considered finding and studying randomly generated Boolean functions (at least with a few inputs and outputs) with good cryptographic properties, to be of high interest.

Based on a algorithm designed by the author which can generate highly nonlinear functions at random, some comparative results are presented that give an insight to differences between constructed and generated Boolean function with good cryptographic properties.

Particular emphasis of the paper is on resiliency of highly nonlinear functions. The random generation algorithm manages to output balanced functions which in some cases have the highest achievable nonlinearity for a particular number of variables and/or have higher nonlinearity then some of the modern methods for obtaining cryptographically strong Boolean functions.

The paper is organized as follows. Section 2 provides some basic definitions and notations that are used throughout the remainder of the article. In Section 3 a random function generator is described, which is used as a foundation for obtaining highly nonlinear resilient functions. Experimental results and comparisons to other research are given in Section 4. Then conclusions follow in Section 5.

2. Preliminaries

We use square brackets to denote vectors like $[a_1,\ldots,a_n]$ and round brackets to denote functions like $f(x_1,\ldots,x_n)$.

2.1. Boolean function

Let $GF(2)=\langle\sum,\oplus,\bullet\rangle$ be two-element Galois field, where $\sum=\{0,1\}$, $\oplus$ and $\bullet$ denotes the sum and multiplication mod 2, respectively. A function $f:\sum^n\mapsto\sum$ is an n-argument Boolean function. Let $z=x_1\cdot 2^{n-1}+x_2\cdot 2^{n-2}+\ldots+x_n\cdot 2^0$ be the decimal representation of arguments $(x_1,x_2,\ldots,x_n)$ of the function f. Let us denote $f(x_1,x_2,\ldots,x_n)$ as y_z. Then $[y_0,y_1,\ldots,y_{2^n-1}]$ is called a truth table of the function f.

2.2. Linear and nonlinear Boolean functions

An n-argument Boolean function f is linear if it can be represented in the following form: $f(x_1,x_2,\ldots,x_n)=a_1x_1\oplus a_2x_2\oplus\ldots\oplus a_nx_n$. Let L_n be a set of all n-argument linear Boolean functions. Let $M_n=\{g:\sum^n\mapsto\sum\mid g(x_1,x_2,\ldots,x_n)=1\oplus f(x_1,x_2,\ldots,x_n)$ and $f\in L_n\}$. A set $A_n=L_n\cup M_n$ is called a set of n-argument affine Boolean functions. A Boolean function $f:\sum^n\mapsto\sum$ that is not affine is called a nonlinear Boolean function.

2.3. Balance

Let $N_0[y_0,y_1,\ldots,y_{2^n-1}]$ be a number of zeros (0's) in the truth table $[y_0,y_1,\ldots,y_{2^n-1}]$ of function f, and $N_1[y_0,y_1,\ldots,y_{2^n-1}]$ be a number of ones (1's). A Boolean function is balanced if

$$N_0[y_0,y_1,\ldots,y_{2^n-1}]=N_1[y_0,y_1,\ldots,y_{2^n-1}].$$

2.4. Algebraic normal form

A Boolean function can also be represented as a maximum of 2^n coefficients of the algebraic normal form (ANF). These coefficients provide a formula for the evaluation of the function for any given input $x=[x_1,x_2,\ldots,x_n]$:

$$f(x)=a_0\oplus\sum_{i=1}^{n}a_ix_i\oplus\sum_{1\le i<j\le n}a_{ij}x_ix_j\oplus\ldots\oplus a_{12\ldots n}x_1x_2\ldots x_n,$$

where $\sum$, $\oplus$ denote modulo 2 summation.

The order of nonlinearity of a Boolean function $f(x)$ is a maximum number of variables in a product term with non-zero coefficient a_J, where J is a subset of $\{1,2,3,\ldots,n\}$. In the case where J is an empty set the coefficient is denoted as a_0 and is called a zero order coefficient. Coefficients of order 1 are $a_1,a_2,\ldots,a_n$, coefficients of order 2 are $a_{12},a_{13},\ldots,a_{(n-1)n}$, coefficient of order n is $a_{12\ldots n}$. The number of all ANF coefficients equals 2^n.

Let us denote the number of all (zero and non-zero) coefficients of order i of function f as $\sigma_i(f)$. For n-argument function f there are as many coefficients of a given order as there are i-element combinations in n-element set, i.e., $\sigma_i(f)=\binom{n}{i}$.

2.5. Hamming distance

Hamming weight of a binary vector $x\in\sum^n$, denoted as $hwt(x)$, is the number of ones in that vector.

Hamming distance between two Boolean functions $f,g:\sum^n\mapsto\sum$ is denoted by $d(f,g)$ and is defined as follows:

$$d(f,g)=\sum_{x\in\sum^n}f(x)\oplus g(x).$$

The distance of a Boolean function f from a set of n-argument Boolean functions X_n is defined as follows:

$$\delta(f)=\min_{g\in X_n}d(f,g),$$

where $d(f,g)$ is the Hamming distance between functions f and g. The distance of a function f a set of affine functions A_n is the distance of function f from the nearest function $g\in A_n$.

The distance of function f from a set of all affine functions is called the nonlinearity of function f and is denoted by N_f.

2.6. Bent functions

A Boolean function $f:\sum^n\mapsto\sum$ is perfectly nonlinear if and only if $f(x)\oplus f(x\oplus\alpha)$ is balanced for any $\alpha\in\sum^n$ such that $1\le hwt(\alpha)\le n$.

For a perfectly nonlinear Boolean function, any change of inputs causes the change of the output with probability of 0.5.

Meier and Staffelbach [24] proved that the set of perfectly nonlinear Boolean functions is the same as the set of Boolean bent functions defined by Rothaus [29].

Perfectly nonlinear functions (or bent functions) have the same, and the maximum possible distance to all affine functions.

Bent functions are not balanced. Hamming weight of a bent function equals $2^{n-1}\pm 2^{\frac{n}{2}-1}$.

2.7. Walsh transform

Let $x=(x_1,x_2,\ldots,x_n)$ and $\omega=(\omega_1,\omega_2,\ldots,\omega_n)$ both belong to $\{0,1\}^n$ and $x\bullet\omega=x_1\omega_1,x_2\omega_2,\ldots,x_n\omega_n$. Let $f(x)$ be a Boolean functions on n variables. Then the Walsh transform of $f(x)$ is a real valued function over $\{0,1\}^n$ that can be defined as:

$$W_f(\omega)=\sum_{x\in\{0,1\}^n}(-1)^{f(x)\oplus x\omega}.$$

The Walsh transform is sometimes called the spectral distribution or simply the spectra of a Boolean function. It is an important tool for the analysis of Boolean function.

2.8. Correlation immunity and resilience

Guo-Zhen and Massey [13] have provided a spectral characterisation of correlation immune functions using Walsh

transform. We can use that as a definition of correlation immunity:

A function $f(x_1, x_2, \ldots, x_n)$ is m-order correlation immune (CI) iff its Walsh transform W_f satisfies $W_f = 0$, for $1 \leq hwt(\omega) \leq m$. Note that balanced m-order correlation immune functions are called m-resilient functions and if f is balanced then $W_f(0) = 0$. Thus, a function $f(x_1, x_2, \ldots, x_n)$ is m-resilient iff its Walsh transform W_f satisfies $W_f(\omega) = 0$, for $0 \leq hwt(\omega) \leq m$.

By an (n, m, d, x) function we mean an n-variable, m-resilient (balanced m-order CI) function with degree d and nonlinearity x. In the above notation the degree component is replaced by a '-' (i.e., $(n, m, -, x)$), if we do not want to specify a degree.

3. Random generation of highly nonlinear functions

As already mentioned earlier, so called bent Boolean functions achieve the highest possible nonlinearity. There exist a number of algorithms for constructing bent Boolean functions. Such constructions have been given by Rothaus [29], Kam and Davida [15], Maiorana [17], Adams and Tavares [1], and others.

Most of the known bent function constructions take bent functions of n arguments as their input and generate bent functions of $n+2$ arguments. One major drawback of these methods is the fact that they are deterministic. Only short bent functions ($n = 4$ or 6) are selected at random and the resulting function is obtained using the same, deterministic formula every time. The possible drawback of such approach (constructions) were stated in the beginning of this paper.

Drawing bent functions at random is not feasible already for small number of arguments ($n > 6$). To make such generation possible, an algorithm was designed that generates random Boolean functions in algebraic normal form thus making use of some basic properties of bent functions to considerably narrow the search space. This makes the generation of bent functions feasible for $n > 6$.

The algorithm for the generation of bent functions in ANF domain takes as its input the minimum and maximum number of ANF coefficients of every order that the resulting functions are allowed to have. Since the nonlinear order of bent functions is less or equal to $n/2$, clearly in ANF of a bent function can not be any ANF coefficient of order higher then $n/2$. This restriction is the major reason for random generation feasibility, since it considerably reduces the possible search space.

However the fact that bent functions are not balanced prohibits their direct application in the cipher system. Still, as bent functions achieve maximum possible nonlinearity they are often used as a foundation for constructing highly nonlinear balanced functions. In recent years some methods have been proposed that transform bent functions to balanced Boolean functions with minimal loss in nonlinearity. Examples of such methods are given in [18] and [19]. Still, balancing bent function can lead to low order of resiliency. In a quest for a randomly generated, highly nonlinear function with higher order resiliency the above mentioned random bent function generation algorithm has been modified to generate such functions. Here again some specific properties of resilient functions are crucial.

As already stated there are certain trade-offs involved among the parameters of a cryptographically sound Boolean function. As it has been showed by Siegenthaler [32] for an n-variable function, of degree d and order of correlation immunity m the following holds: $m + d \leq n$. Further, if the function is balanced then $m + d \leq n - 1$.

The generating algorithm is used basically in the same way as when generating bent functions. Still it operates in the ANF domain and it takes as its input the number minimal and maximal number of coefficients of every order. Nonlinear order is restricted according to Siegenthalter's findings and some more precise upper bounds on resilient order given by Sarkar and Maitra in [30].

Sarkar and Maitra in [30] present some construction methods for highly nonlinear resilient functions and give upper bounds on nonlinearity of resilient functions.

For the sake of completeness a Maiorana-McFarland like construction technique will now be briefly discussed. This technique is perhaps the most important of all resilient Boolean functions construction methods and has been investigated in a number of papers [2, 3, 5, 31]. This construction has been used by Maitra and Sarkar as a basis for their work.

Let π be a map from $\{0,1\}^r$ to $\{0,1\}^k$, where for any $x \in \{0,1\}^r$, $hwt(\pi(x)) \geq m+1$. Let $f : \{0,1\}^{r+k} \mapsto \{0,1\}$ be a Boolean function defined as $f(x,y) = y \bullet \pi(x) \oplus g(x)$, where $x \in \{0,1\}^r$, $y \in \{0,1\}^k$ and $y \bullet \pi(x)$ is the inner product of y and $\pi(x)$. Then f is m-resilient.

Table 1
Upper bounds on nonlinearity of resilient functions

	5	*6*	*7*	*8*	*9*	*10*
1	12	24	56	116*	244*	492*
2	8	24	56*	112	240	480
3	0	16	48	112	240*	480
4		0	32	96	224	480*
5			0	64	192	448
6				0	128	384
7					0	256
8						0

Table 1 summarises the results obtained in [30] and gives upper bounds on nonlinearity of resilient functions for number of arguments ranging from 5 to 10. The rows represent the resiliency and the columns represent the number of variables. Entries with * indicate bounds which have not yet been achieved. Functions can be constructed with parameters satisfying the other entries.

Table 1 can be used as a benchmark for assessing the efficacy of resilient functions construction methods.

4. Experimental results

Now let's see the results from above mentioned random resilient function generator against the upper bounds presented in Table 1.
The maximum nonlinearity is known for all Boolean functions on even number of variables – it is achieved by bent functions. The maximum nonlinearity for odd variable Boolean functions is known for $n \leq 7$. Also, maximum nonlinearity question is solved for balanced and resilient functions on n variables for $n \leq 5$ (which is easy to do by exhaustive computer search). Let's consider cases for $6 \leq n \leq 10$.

- $n = 6$: Maximum nonlinearity for $n = 6$ is 28 (for bent functions). Maximum nonlinearity of a balanced function is 26 and construction of such functions is known. Maximum nonlinearities for 1, 2 and 3-resilient functions were shown (be computer search) to be 24, 24 and 16. Random resilient function generator presented in this paper is able to generate 1, 2 and 3-resilient functions.

- $n = 7$: Maximum nonlinearity of a balanced Boolean functions for $n = 7$ is 56. As shown in [30] the maximum nonlinearities for 1, 2, 3 and 4-resilient functions are respectively 56, 56, 48, 32. However 2-resilient function with nonlinearity of 56 is not known. Random generator is able to generate all these resilient functions except that (7,2,-,56).

- $n = 8$: Nonlinearity of 8 argument bent function is 120. Maximum (theoretical) nonlinearity for a balanced function is 118, however such function if not known. Maximum possible nonlinearities for 1, 2, 3, 4 and 5-resilient functions are 116, 112, 112, 96, and 64. The existence of (8,1,-,116) function is an open problem. Constructions for other functions are known. Random generator can output all the functions except the not known (8,1,-,116) and (8,3,-,112).

- $n = 9$: Maximum nonlinearity of such functions as an open problem. The known upper bound if 244. It is easy to construct a function with nonlinearity of 240. Maximum nonlinearities of resilient functions are 244, 240, 240, 224, 192, 128 for 1, 2, 3, 4, 5, 6-resilient functions respectively. The generator is capable of generating (9,1,-,240), (9,2,-,224), (9,5,-,192) and (9,6,-,128) functions.

- $n = 10$: The nonlinearity of a bent function is 496. Maximum nonlinearity of a balanced function is 494, best know function has linearity of 492. 492, 488, 480, 480, 448, 384, 256 are the nonlinearities of 1,2, 3, 4, 5, 6, 7-resilient function. Constructions of the following functions are not known: (10,1,-,492), (10,1,-,488), (10,2,-,488), (10,4,-,480). Random generator can generate the following: (10,1,-,480), (10,3,-,448), (10,5,-,384), (10,7,-,256).

5. Conclusions

As shown in the previous paragraph, the random resilient function generator is capable of generating Boolean functions having some very promising cryptographic qualities. In many cases these functions are on par with the best known constructions. In other cases they fall slightly short of best achievable results. In any case they have the advantage of being truly random and not being restricted by specific constraints associated with each specific design. One can suspect that such constraints may render the function (or a cipher system based on it) vulnerable to some future cryptographic attack.
Also, results presented in this article are the very first results from the resilient function generator. It's output relies heavily on parameter setting, mainly on the number of higher order ANF coefficients in the resulting function. As this dependencies are investigated we might expect still better results from the generator.
As with generated bent functions, also generated resilient functions can have a very compact (small) algebraic normal form which can be utilized for efficient storage and fast cryptographic routines.

Acknowledgements

This scientific paper has been financed as a research project from Polish State Committee For Scientific Research funds in the years 2004–2006.

References

[1] C. M. Adams and S. E. Tavares, "Generating and counting binary bent sequences", *IEEE Trans. Inform. Theory*, vol. IT-36, pp. 1170–1173, 1990.

[2] P. Camion, C. Carlet, P. Charpin, and N. Sendrier, "On correlation immune functions", in *Adv. Crypt. CRYPTO 1991*, Santa Barbara, USA, 1991, pp. 86–100.

[3] C. Carlet, "More correlation immune and resilient functions over Galois fields and Galois rings", in *Adv. Crypt. EUROCRYPT 1997*, Konstanz, Germany, 1997, pp. 422–433.

[4] C. Carlet, "On the coset weight divisibility and nonlinearity of resilient and correlation immune functions", in *SETA 2001*, Bergen, Norway, 2001.

[5] S. Chee, S. Lee, D. Lee, and S. H. Sung, "On the correlation immune functions and their nonlinearity", in *Advances in Cryptology ASIACRYPT 1996*, *LNCS*. Berlin: Springer, 1996, vol. 1163, pp. 232–243.

[6] J. A. Clark and J. L. Jacob, "Two stage optimisation in the design of Boolean functions", in *5th Australasian Conference on Information Security and Privacy, ACISP 2000*, E. Dawson, A. Clark, and C. Boyd, Eds., *LNCS*. Berlin: Springer, 2000, vol. 1841, pp. 242–254.

[7] J. A. Clark, J. L. Jacob, and S. Stepney, "The design of S-boxes by simulated annealing", in *Int. Conf. Evol. Comput. CEC 2004*, Portland, USA, 2004.

[8] J. A. Clark, J. L. Jacob, and S. Stepney, "Secret agents leave big footprints: how to plant a cryptographic trapdoor, and why you might not get away with it", in *Genetic and Evolutionary Computation Conference GECCO 2003*, *LNCS*. Berlin: Springer, 2003, vol. 2724, pp. 2022–2033.

[9] J. A. Clark, J. L. Jacob, and S. Stepney, "Functions satisfying multiple criteria", in *Progress in Cryptology INDOCRYPT 2002*, *LNCS*. Berlin: Springer, 2002, vol. 2551, pp. 246–259.

[10] J. A. Clark, J. L. Jacob, and S. Stepney, "Searching for cost functions", in *Int. Conf. Evol. Comput. CEC 2004*, Portland, USA, 2004, pp. 1517–1524.

[11] H. Dobbertin, "Construction of bent functions and balanced functions with high nonlinearity", in *Fast Software Encryption, 1994 Leuven Workshop*, *LNCS*. Berlin: Springer, 1994, vol. 1008, pp. 61–74.

[12] R. Forré, "The strict avalanche criterion: spectral properties of Boolean functions with high nonlinearity", in *Advances in Cryptology: CRYPTO 1988*, *LNCS*. Berlin: Springer, 1990, vol. 403, pp. 450–468.

[13] X. Guo-Zhen and J. Massey, "A spectral characterization of correlation immune combining functions", *IEEE Trans. Inform. Theory*, vol. 34, no. 3, pp. 569–571, 1988.

[14] X. D. Hou, "On the norm and covering radius of first-order Reed-Muller codes", *IEEE Trans. Inform. Theory*, vol. 43, no. 3, pp. 1025–1027, 1997.

[15] J. B. Kam and G. Davida, "Structured design of substitution-permutation encryption networks", *IEEE Trans. Comput.*, vol. C-28, pp. 747–753, 1979.

[16] K. Kurosawa and T. Satoh, "Generalization of higher order SAC to vector output Boolean functions", *IEICE Trans.*, vol. E90, no. 1, 1998.

[17] J. A. Maiorana, "A class of bent functions", R41 Tech. Paper, 1971.

[18] S. Maity and T. Johansson, "Construction of cryptographically important Boolean functions", in *INDOCRYPT 2002*, Hyderabad, India, 2002, pp. 234–245.

[19] S. Maity and S. Maitra, "Minimum distance between bent and 1-resilient Boolean functions", in *FSE 2004*, New Delhi, India, 2004, pp. 143–160.

[20] S. Maitra, "Highly nonlinear balanced Boolean functions with very good autocorrelation property", Tech. Rep. 2000/047, Indian Statistical Institute, Calcutta, 2000.

[21] S. Maitra, "Autocorrelation properties of correlation immune Boolean functions", in *INDOCRYPT 2001*, Chennai, India, 2001, pp. 242–253.

[22] S. Maitra and E. Pasalic, "Further constructions of resilient Boolean functions with very high nonlinearity", in *SETA 2001*, Bergen, Norway, 2001.

[23] M. Matsui, "Linear cryptanalysis method for DES cipher (abstracts)", in *EUROCRYPT 1993*, Lofthus, Norway, 1993.

[24] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions", in *Advances in Cryptology: EUROCRYPT 1989*, J. J. Quisquater, J. Vandewalle, Eds., *LNCS*. Berlin: Springer, 1989, vol. 434, pp. 549–562.

[25] W. Millan, A. Clark, and E. Dawson, "Heuristic design of cryptographically strong balanced Boolean functions", in *Advances in Cryptology: EUROCRYPT 1998*, *LNCS*. Berlin: Springer, 1998, vol. 1403, pp. 489–499.

[26] K. Nyberg, "Perfect nonlinear S-boxes", in *Advances of Cryptology: EUROCRYPT 1991*, *LNCS*. Berlin: Springer, 1991, vol. 547, pp. 378–386.

[27] E. Pasalic, S. Maitra, T. Johansson, and P. Sarkar, "New constructions of resilient and correlation immune Boolean functions achieving upper bound on nonlinearity", in *Workshop on Coding Theory, Electronic Notes in Discrete Mathematics*, Elsevier, 2001.

[28] B. Preneel, W. Van Leekwijck, L. Van Linden, R. Govaerts, and J. Vandewalle, "Propagation characteristics of Boolean functions", in *Advances in Cryptology: EUROCRYPT 1990*, *LNCS*. Berlin: Springer, 1991, vol. 473, pp. 161–173.

[29] O. S. Rothaus, "On bent functions", *J. Combin. Theory: Ser. A*, vol. 20, pp. 300–305, 1976.

[30] P. Sarkar and S. Maitra, "New directions in design of resilient Boolean functions", Tech. Rep. ASD/2000/04, Indian Statistical Institute, 2000.

[31] J. Seberry, X. M. Zhang, and Y. Zheng, "On constructions and nonlinearity of correlation immune Boolean functions", in *Advances in Cryptology: EUROCRYPT 1993*, Lofthus, Norway, 1994, pp. 181–199.

[32] T. Siegenthaler, "Correlation-immunity of nonlinear combining functions for cryptographic applications", *IEEE Trans. Inform. Theory*, vol. IT-30, no. 5, pp. 776–780, 1984.

[33] J. J. Son, J. I. Lim, S. Chee, and S. H. Sung, "Global avalanche characteristics and nonlinearity of balanced Boolean functions", *Inform. Proces. Lett.*, vol. 65, no. 3, pp. 139–144, 1998.

[34] S. H. Sung, S. Chee, and C. Park, "Global avalanche characteristics and propagation criterion of balanced Boolean functions", *Inform. Proces. Lett.*, vol. 69, no. 1, pp. 21–24, 1999.

[35] Y. Tarannikov, "On resilient Boolean functions with maximal possible nonlinearity", Tech. Rep. 2000/005, Mech. & Math. Department, Moscow State University, 2000.

[36] X. M. Zhang and Y. Zheng, "GAC – the criterion for global avalanche characteristics of cryptographic functions", *J. Univ. Comput. Sci.*, vol. 1, no. 5, pp. 316–333, 1995.

Anna Grocholewska-Czuryło is an adjunct at the Poznań University of Technology. She has studied and published papers on a range of topics like natural language processing, cellular automata, neural networks and has finally focused on data security and cryptology, especially methods of designing Boolean functions and s-box design. She has earned her Ph.D. degree entitled: "Pseudobalanced and Bent Boolean Functions and Algorithms of their Generating Methods" in 2001.

e-mail: czurylo@sk-kari.put.poznan.pl
Institute of Control and Information Engineering
Poznań University of Technology
Marii Skłodowskiej-Curie Sq. 5
60-965 Poznań, Poland

Paper

End-to-end service survivability under attacks on networks

Wojciech Molisz and Jacek Rak

Abstract— Network survivability is a capability of a networked system to provide its services despite failures or attacks. Attacks, e.g., due to acts of war, being potentially damaging events, were basically considered in the historical definitions of a survivability phenomenon. The meaning of the term: "network survivability" evolved in the last decade. Recently, attacks replayed the important role again. Their nature, however, including intrusions, probes, denials of service, differs from the old one. Survivability is strongly related to other fields of study. In particular, quality of service depends on network survivability. We investigate these dependencies in scale-free networks. Many networks are scale-free, i.e., their node degree distribution follows the power law. Nodes of the highest degrees, called centers, are highly vulnerable to attacks. Elimination of these nodes seriously degrades the overall performance of network services. In this paper we propose a model, which, based on traffic parameters of a demand, like delay or bit rate, allows to establish the survivable and attack proof end-to-end connections. The key idea of this model is that for the significant traffic, it establishes paths, which omit centers. The important connections become more resistant to attacks. We show that in the best case, obtained for the highest class of service, the number of broken connections is reduced even by factor 3. Example results are compared to those for the standard distance metrics. Our model is applicable to many network architectures and many classes of service.

Keywords— survivable data networks, attacks on networks, scale-free networks, routing, resource allocation.

1. Introduction

Network survivability is a capability of a networked system to provide its services despite failures or attacks. Attacks, e.g., due to acts of war, being potentially damaging events, were basically considered in the historical definitions of a survivability phenomenon. In the last decade, focus was rather on protecting systems against failures, due to software defects, hardware faults or human errors. Recently, attacks replayed the important role again. Their nature, however, including intrusions, probes, denials of service, differs from the old one.

Survivability is strongly related to other fields of study, including fault-tolerance, reliability, safety or performance evaluation. In particular, quality of service depends on network survivability. In this paper we investigate these dependencies in scale-free networks (SFNs). Many networks are scale-free, i.e., their node degree distribution follows the power law. Nodes of the highest degrees, called centers, are highly vulnerable to attacks. Elimination of these nodes seriously degrades the overall performance of network services.

Redundancy is the key to provide services in the face of attacks or failures. Survivability, based on the experience of fault tolerance, assumes various techniques of protection and restoration. Protection in our model is based on the pre-planned backup path for each active (working) end-to-end path [7, 10]. Depending on allowable costs, protecting paths may be either dedicated or shared. This is typical for all survivable networks. The scale-free networks, however, need a special treatment.

Centers in such networks are connected to many other nodes by links of high capacities, switch large amount of data and are of great degree. They are excellent goals of malicious attacks, performed by intruders getting the maximum destructive effect at minimum cost.

An example of a scale-free network is shown in Fig. 1.

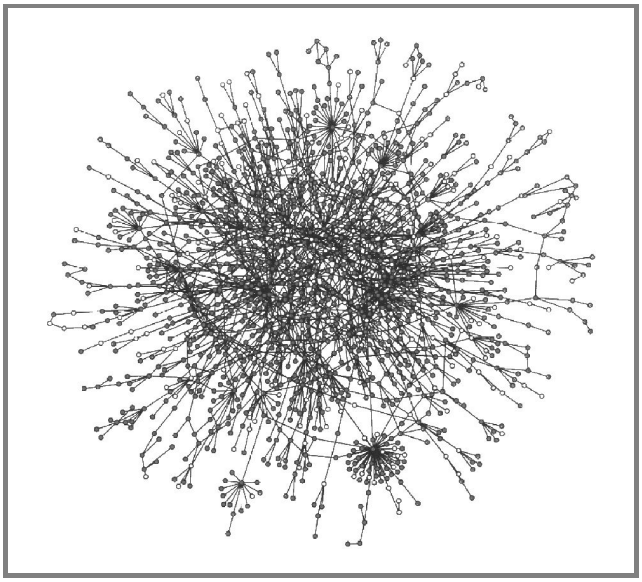


Fig. 1. An example of a scale-free network.

Centers exist in many networks. It has been proved, that the uncontrolled growth of a network leads to a power law distribution of node degrees ($P(k) \sim k^{-\gamma}$) [2]. Figure 2 presents the degree distribution $P(k)$ of scale-free networks compared to random topologies (having the Poisson degree distribution).

By uncontrolled growth we mean adding the new elements according to the preferential attachment rule. Following such a rule, network nodes are mostly being attached to the already highly connected ones. This phenomenon, often

referred to as the *rich get richer process*, causes networks to become scale-free.

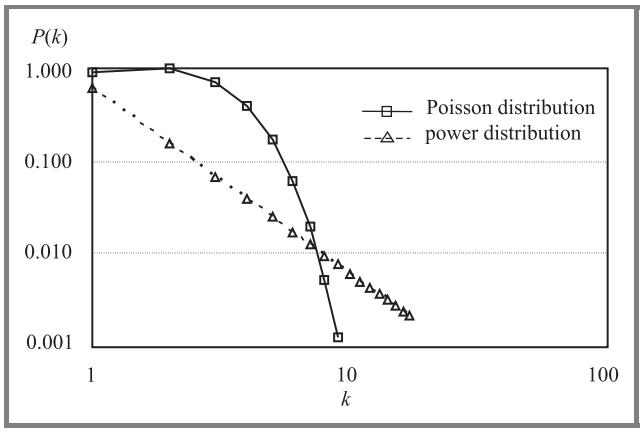


Fig. 2. Comparison of Poisson and power law distributions.

It has been proved that the topology of many wide-area networks is scale-free [1, 4, 6, 8]. Such topologies can be also found in other fields of science, for instance in chemistry or biology. Some examples of power law distribution that are worth mentioning can be found for the stock-price fluctuations, movie actor collaboration networks, biological cellular networks, scientific citation networks and many others.

If connections are established with help of the shortest path algorithm and the standard distance metrics, nodes of extraordinary high degree are often the transit elements of the installed paths. We define the parameter named *betweenness centrality* with regard to a certain node k, denoted $BC(k)$ as [5]:

$$BC(k) = \sum_{p \neq q} \frac{\delta_k(p,q)}{\delta(p,q)}, \tag{1}$$

where:

$\delta(p,q)$ is the number of shortest paths between nodes p and q,

$\delta_k(p,q)$ is the number of shortest paths between nodes p and q that run through a vertex k.

Each central node has a high value of BC, because many shortest paths established between various pairs of nodes (p,q) traverse such nodes. Attacks on centers may thus cause many connections to fail.

In this paper we propose a model, which, based on traffic parameters of a demand, allows to establish the survivable end-to-end paths in scale-free networks. Specific properties of scale-free networks imply the choice of a new metrics of the cost of a link and new algorithms finding paths, reducing the number of connections broken due to an attack.

The rest of the paper is organized as follows. Section 2 presents the model of establishing survivable connections with regard to the requested level of continuity of service in scale-free networks. Section 3 presents the integer linear programming (ILP) formulation of the problem and the corresponding heuristic algorithm. Modeling results, obtained for an artificial scale-free network, are shown in Section 4 and include: the probability of demand rejection, length of active path, the number of broken connections and the values of restoration time.

2. Model description

The main objective of our model is to establish survivable connections in the way that the number of connections broken due to an attack on a node is reduced. The decrease in number of broken connections is to be greater with the increase of the required level of service. It can be achieved with help of a metrics that, depending on the requested class of service, forces active paths to bypass nodes of high degree.

We consider the network consisting of N nodes and A directed arcs. Nodes and arcs are numbered:

$n-1, \ldots, N$;

$a = 1, \ldots, A$.

We assume $K \leq N(N-1)$ demands. We denote:

$p_k(q_k)$	source (destination) of a demand k; $k = 1, 2, \ldots, K$
c_a	capacity of an arc a
Λ_a	number of channels of an arc a
$\prod_{p,q}$	an active path from p_k to q_k
$\overline{\prod}_{p,q}$	a backup path from p_k to q_k
$c\left[\prod_{p,q}\right]$	the requested capacity for a connection from p_k to q_k
$\overline{C}_a$	spare capacity on an arc a

M classes of service are assumed, numbered from 0 to $M-1$. Class 0 represents demands for which the probability of breaking the connections must be minimized. With the increase of a class number, the requested level of service continuity gets lower. Various types of traffic of many network architectures (e.g., ATM) can be mapped onto the proposed classes, which makes the approach widely applicable.

In order to find active paths on a shortest-path basis, we propose the metrics that determines the cost κ_a a of an arc a, as shown in Eq. (2):

$$\kappa_a^m = \begin{cases} \dfrac{m}{M-1} d_a^* + \dfrac{(M-1)-m}{M-1} BC^*(n) & \text{if } c\left[\prod_{p,q}\right] \leq \overline{C}_a \\ \infty & \text{otherwise} \end{cases}, \tag{2}$$

where:

m is the current class of service; $m = 0, 1, 2, \ldots, M-1$,

d_a^* is the normalized length of an arc a:

$$d_a^* = \frac{d_a}{\max\limits_a (d_a)}. \tag{3}$$

$BC^*(n)$ is the normalized value of the betweenness centrality of a node n:

$$BC^*(n) = \frac{BC(n)}{\max\limits_{n}\left(BC(n)\right)}. \tag{4}$$

If there is not enough spare capacity on an arc a to install the active path (according to the requested capacity $c[\prod_{p,q}]$), then the cost of using this arc is set to infinity. Otherwise the cost is the weighted sum of the normalized length of an arc (d_a^*) and the value of the normalized betweenness centrality coefficient $BC^*(n)$ of the end node n of an arc a. Two boundary cases are worth explanation:

- Class 0 for demands of the highest quality of service. Here the cost of the arc is calculated only on the basis of the value of $BC^*(n)$. This results in installing paths that omit central nodes. This causes the connections of class 0 to have a low probability of breaking. However, the installed active paths are not the shortest ones.

- Class $M-1$ for connections that do not require the guarantee of continuity. For them the cost of each arc is determined by Eq. (2) and is thus equal to:

$$\kappa_a^{M-1} = \begin{cases} d_a^* & \text{if } c\left[\prod_{p,q}\right] \le \overline{C}_a \\ \infty & \text{otherwise} \end{cases}. \tag{5}$$

  Here, the active paths of connections are the shortest ones in the sense of distance but often run through central nodes and are thus exposed to attacks.

For classes $1,2,\ldots,M-2$ characteristics of active paths with regard to their length and vulnerability to attacks are expected to be the compromise of the corresponding features of classes 0 and $M-1$, respectively.
Backup paths are found using the standard distance metrics. In order to allow fast restoration of connections of each class in case of a failure, the backups are computed as the shortest ones.

3. Methods used to compute active and backup paths

In the following part of the paper, the protection against a single node failure is assumed. All backup paths are dedicated (not shared). There is only one backup path for each connection (path protection model).

3.1. The ILP formulation of the problem

We propose using the following nomenclature:

N set of nodes of a network

A set of directed link (arcs) in a network

K number of source-destination (demand) pairs of nodes, $K \le N(N-1)$

p_k source node of a demand k

q_k destination node of a demand k

$\alpha_{k,a}^{\lambda}$ takes value of 1 if a channel λ of an arc a is used by an active path of a demand k; 0 – otherwise

$\beta_{k,a}^{\lambda}$ takes value of 1 if a channel λ of an arc a is used by a backup path of a demand k; 0 – otherwise

λ_a a capacity of an arc a represented by the number of channels: $\forall_a \lambda_a = \Lambda$

$\kappa_{k,a}^{m}$ the channel cost in an arc a calculated by considering the class m of a demand k and the length of an arc according to Eq. (2) (for active paths)

$s_{k,a}$ the channel cost in an arc a calculated for a demand k with regard to its length (for backup paths)

x vector of all components of flows (variables)

It is to find paths transporting required flows from sources to destinations, protecting them against a single node failure and minimizing the linear cost:

$$\varphi(x) = \sum_{k=1}^{K}\sum_{a=1}^{A}\sum_{\lambda=1}^{\lambda_a}\left(\kappa_{k,a}^{\lambda}\cdot\alpha_{k,a}^{\lambda} + s_{k,a}\cdot\beta_{k,a}^{\lambda}\right) \tag{6}$$

subject to constraints given in Eqs. (7)–(15).

a) *Capacity constraints (on the number of the available wavelengths on an arc a)*:

$$\sum_{\lambda=1}^{\lambda_a}\sum_{k=1}^{K}\left(\alpha_{k,a}^{\lambda} + \beta_{k,a}^{\lambda}\right) \le \lambda_a. \tag{7}$$

b) *The flow balance constraints for each wavelength λ and for each demand k*:

For a source node of an active path:

$$\sum_{\{a:a\equiv(p_k,j);j=1,2,K,N;j\neq p_k\}}\alpha_{k,a}^{\lambda} - \sum_{\{a:a\equiv(i,p_k);i=1,2,K,N;i\neq p_k\}}\alpha_{k,a}^{\lambda} = 1. \tag{8}$$

For a destination node of an active path:

$$\sum_{\{a:a\equiv(q_k,j);j=1,2,K,N;j\neq q_k\}}\alpha_{k,a}^{\lambda} - \sum_{\{a:a\equiv(i,q_k);i=1,2,K,N;i\neq q_k\}}\alpha_{k,a}^{\lambda} = -1. \tag{9}$$

For transit nodes of an active path:

$$\sum_{\{a:a\equiv(i,j);j=1,2,K,N;i,j\neq p_k,i,j\neq q_k\}}\alpha_{k,a}^{\lambda} - \sum_{\{a:a\equiv(i,j);i=1,2,K,N;i,j\neq p_k,i,j\neq q_k\}}\alpha_{k,a}^{\lambda} = 0. \tag{10}$$

For a source node of a backup path:

$$\sum_{\{a:a\equiv(p_k,j);j=1,2,K,N;j\neq p_k\}}\beta_{k,a}^{\lambda} - \sum_{\{a:a\equiv(i,p_k);i=1,2,K,N;i\neq p_k\}}\beta_{k,a}^{\lambda} = 1. \tag{11}$$

For a destination node of a backup path:

$$\sum_{\{a:a\equiv(q_k,j);j=1,2,K,N;j\neq q_k\}} \beta_{k,a}^{\lambda} - \sum_{\{a:a\equiv(i,q_k);i=1,2,K,N;i\neq q_k\}} \beta_{k,a}^{\lambda} = -1. \quad (12)$$

For transit nodes of a backup path:

$$\sum_{\{a:a\equiv(i,j);j=1,2,K,N;i,j\neq p_k,i,j\neq q_k\}} \beta_{k,a}^{\lambda} - \sum_{\{a:a\equiv(i,j);i=1,2,K,N;i,j\neq p_k,i,j\neq q_k\}} \beta_{k,a}^{\lambda} = 0. \quad (13)$$

c) *Constraints assuring nodal-disjointness of active and backup paths*:

$$\sum_{\lambda=1}^{\lambda_a} \sum_{\{a:a\equiv(i,j);j=1,2,K,N;j\neq i;i\neq p_k\}} \left(\alpha_{k,a}^{\lambda} + \beta_{k,a}^{\lambda}\right) \leq 1, \quad (14)$$

$$\sum_{\lambda=1}^{\lambda_a} \sum_{\{a:a\equiv(i,j);i=1,2,K,N;i\neq j;j\neq q_k\}} \left(\alpha_{k,a}^{\lambda} + \beta_{k,a}^{\lambda}\right) \leq 1. \quad (15)$$

Constraint, given in Eq. (7), assures that the total number of channels, reserved for survivable connections on an arc a, will not exceed the capacity of this arc. For each channel and each demand, flow balance for the active paths is assured by Eqs. (8)–(10). For instance, Eq. (8) guarantees that there is only one active path outgoing from the source node s_k. Equation (10) simply states that transit nodes do not store traffic. Equations (11)–(13) describe the flow balance constraints for backup paths, respectively.

3.2. The SACC heuristic algorithm

SACC algorithm of establishing survivable and attack-compliant connections

Input:

- A pair of source and destination nodes $[p_k, q_k]$
- Requested capacity $c\,[\prod_{p,q}]$
- Requested class of service m
- Number of classes of service M

1. Find the active path of a connection:
   - 1.1. For each arc a calculate its cost as defined in Eq. (2)
   - 1.2. Find the shortest path between nodes p_k and q_k, using the matrix of costs C, evaluated in 1.1
   - 1.3. If the active path is found then install it, else go to Step 3
2. Find the backup path:
   - 2.1. For each arc a evaluate its cost as defined in Eq. (5)
   - 2.2. In order to assure the nodal disjointness of active and backup paths of a connection, set the costs of active path's arcs as well as arcs incident to active path's nodes to infinity
   - 2.3. Find the shortest path between nodes p_k and q_k, using the matrix of costs C, evaluated in 2.1
   - 2.4. If the backup path is found then install the path
3. If any of the two paths cannot be found due to the lack of spare resources, then reject the demand and remove the active path (if installed), else establish the connection

The main advantage of using heuristic methods over ILP approach is their polynomial complexity. SACC algorithm, described below, uses a Dijkstra's algorithm to find a shortest path between a pair of a source and destination nodes.

4. Modeling results

In this section we focus our research on measuring the probability of demand rejection, the average length of active path, the number of broken connections and restoration times for various classes of service. The scale-free network, shown in Fig. 3, used in research, was generated by Pajek software.

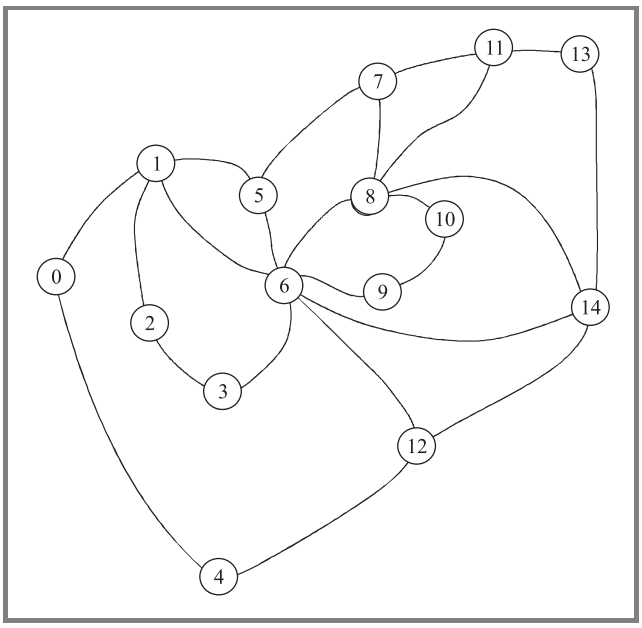


Fig. 3. The SFN artificial scale-free network.

According to the values of the normalized betweenness centrality parameter, given in Table 1, nodes: 1, 6, 8, 14 having high values of BC^* can be referred to as centers.

Table 1
Values of the normalized betweenness centrality for nodes of the SFN network

Node k	0	**1**	2	3	4	5	**6**
$BC^*(k)$	0.063	0.367	0.023	0.077	0.046	0.135	1.000

7	**8**	9	10	11	12	13	**14**
0.085	0.462	0.081	0.035	0.058	0.213	0.027	0.300

All the paths for survivable connections were obtained with help of AMPL/CPLEX environment as well as with dedicated network simulator, implemented in C++ environment.

Due to the complexity of the optimal algorithm (NP completeness of the investigated ILP formulation) calculation of paths was infeasible for real large networks. Simulation focused on measuring the number of broken connections and restoration times was then run with the use of our dedicated network simulator.

The following properties were assumed for all the directed links of the network:

– equal number of available channels (here 8);

– equal capacity of all channels;

– equal length (1890 km).

For each connection we assumed:

– class of service chosen randomly out of M available classes;

– metrics given in Eq. (2) in all active path computations;

– standard distance metrics, given in Eq. (5), in all backup path computations;

– a demand of resource allocation equal to the capacity of one channel of a link;

– protection against a single node failure.

Connections were broken due to attacks on nodes (one each time). The probability of each node to be attacked was assumed to be proportional to the values of $BC^*(k)$. During each experiment for the SFN network, 30 logical topologies were generated. Each logical topology is a graph of a fixed (here 10) number of randomly chosen pairs of source and destination nodes. For each logical topology, failures of 100 nodes were generated to measure the numbers of broken connection and values of restoration time.

4.1. The ILP results

Table 2 shows results of path computation obtained by solving the optimization problem stated in Subsection 3.1 with help of AMPL/CPLEX environment.

It was to find paths for connections of 3 classes of service available. The results prove that smaller the class number was (meaning higher requested level of service continuity), more active paths omitted nodes of high degree. For instance a connection between nodes 1 and 14 of the highest (0) class of service, was realized by establishing a long (but omitting centers; here 6, 8) active path: (1, 5, 7, 11, 13, 14) and a much shorter backup path (1, 6, 14). On the contrary, a connection of the lowest class (here 2), between nodes 0 and 4, was realized by a short active path (0, 4) and a much longer backup path (0, 1, 6, 12, 4).

Table 2

Example paths for 10 survivable connections of 3 classes of service (0, 1 and 2), established according to the proposed model

Connection:	(0,4)		priority:		2	
Active path:	0	4				
Backup path:	0	1	6	12	4	
Connection:	(0,11)		priority:		1	
Active path:	0	1	5	7	11	
Backup path:	0	4	12	6	8	11
Connection:	(1,12)		priority:		1	
Active path:	1	6	12			
Backup path:	1	0	4	12		
Connection:	(1,14)		priority:		0	
Active path:	1	5	7	11	13	14
Backup path:	1	6	14			
Connection:	(2,7)		priority:		1	
Active path:	2	1	5	7		
Backup path:	2	3	6	8	7	
Connection:	(2,11)		priority:		0	
Active path:	2	1	5	7	11	
Backup path:	2	3	6	8	11	
Connection:	(5,11)		priority:		1	
Active path:	5	7	11			
Backup path:	5	6	8	11		
Connection:	(5,13)		priority:		2	
Active path:	5	7	11	13		
Backup path:	5	6	14	13		
Connection:	(6,8)		priority:		2	
Active path:	6	8				
Backup path:	6	14	8			
Connection:	(0,14)		priority:		2	
Active path:	9	6	14			
Backup path:	9	10	8	14		

4.2. Probability of demand rejection

We considered the phenomenon of demand rejection due to all possible reasons. The main causes included:

– lack of available link channels;

– topology bottlenecks.

Figure 4 shows the probability of demand rejection for 3 classes of service, while Fig. 5 – for 6 classes of service, respectively.

Comparing results for M classes of service, we see that CPLEX served all the demands. This is due to the fact that only 10 demands were defined for each logical topology and there was enough resources to serve all of them. In contrast, our heuristic algorithm found local optima and,

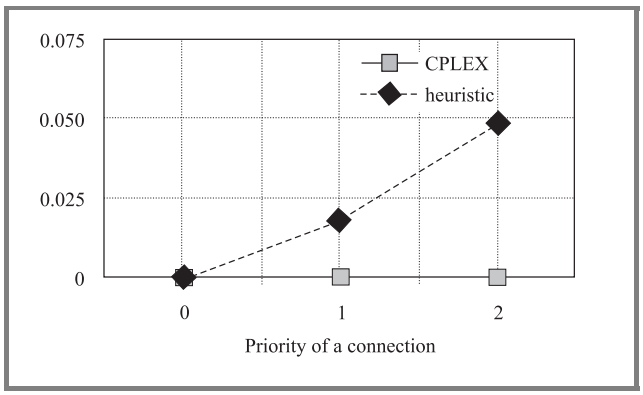


Fig. 4. Probability of demand rejection for 3 classes of service.

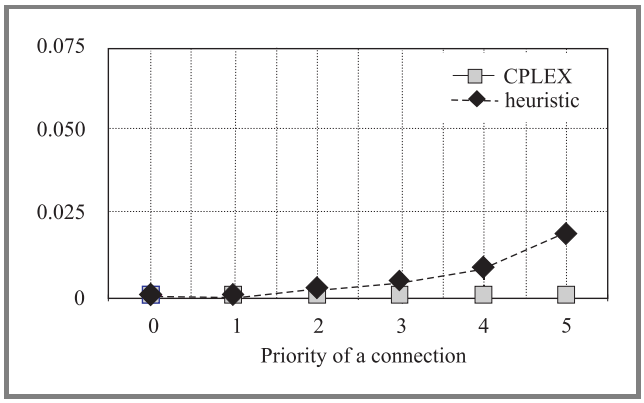


Fig. 5. Probability of demand rejection for 6 classes of service.

due to bottlenecks, some of the demands were rejected. There is no significant difference in the total number of rejected demands between 3 and 6 classes of service.

4.3. Number of active path links

Figures 6 and 7, and Table 3 show the numbers of active path links as the function of the connection class of service. When increasing the level of the requested continuity of service of a connection (decreasing the class number), its active path becomes longer. This is because it omits nodes

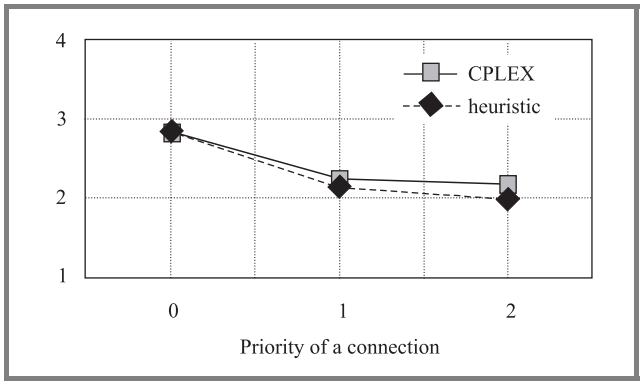


Fig. 6. Average number of active path links for 3 classes of service.

of high degree. In the worst case for 3 classes of service, the average length of active path obtained for the class 0, was about 1.31 times worse than for the class 2.

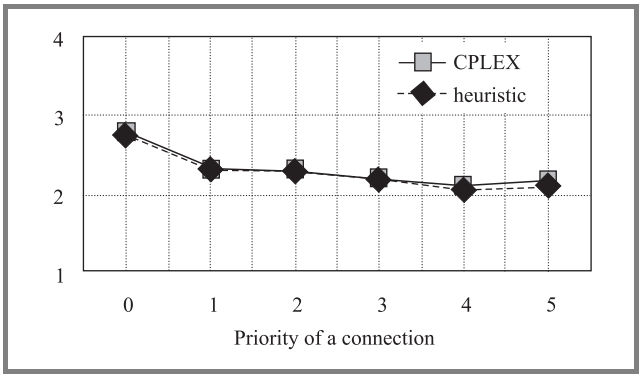


Fig. 7. Average number of active path links for 6 classes of service.

Table 3
Average number of active path links for 3 and 6 classes of service (CPLEX)

Number of service classes 3				
Class number		0	1	2
Active path length [links]	(CPLEX)	2.84	2.23	2.16
	(heuristics)	2.84	2.15	1.98
95% confidence interval [links]	(CPLEX)	0.48	0.27	0.27
	(heuristics)	0.49	0.26	0.26

Number of service classes 6							
Class number		0	1	2	3	4	5
Active path length [links]	(CPLEX)	2.79	2.32	2.33	2.22	2.12	2.19
	(heuristics)	2.77	2.32	2.30	2.20	2.06	2.09
95% confidence interval [links]	(CPLEX)	0.69	0.50	0.46	0.38	0.37	0.33
	(heuristics)	0.67	0.51	0.46	0.39	0.38	0.33

Typically, heuristic algorithms give worse results than the respective ILP ones. Generally, this remains true for our SACC algorithm. However, due to the non-zero probability of rejecting the demands for the SACC algorithm (Figs. 4 and 5), the obtained active paths turned out to be shorter than the respective ILP ones. This was due to the smaller network congestion, obtained when SACC algorithm was used, as it established less connections than the ILP algorithm. This feature caused similar effect regarding the aggregate number of broken connections and the total restoration time, described in the next subsections.

4.4. Number of broken connections

Figures 8 and 9 show the aggregate numbers of broken connections as the function of the connection class of service. They prove that the proposed model results in a significant decrease in the number of broken connections. The higher the requested level of service continuity is, the decrease

in the number of broken connections gets more visible. In the best case, observed for 6 classes of service, about 67% less connections were broken for the class 0, compared to the results for the class 5.

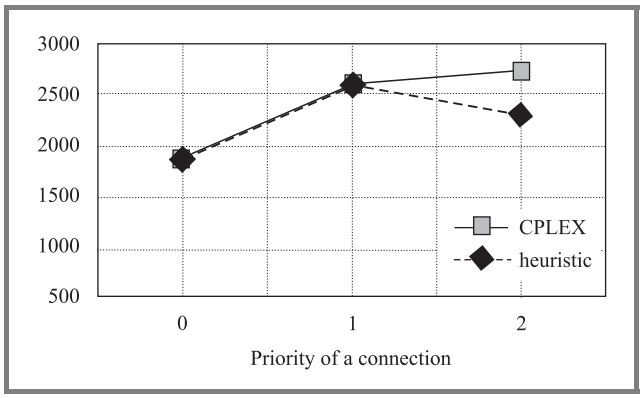


Fig. 8. Aggregate number of broken connections for 3 classes of service.

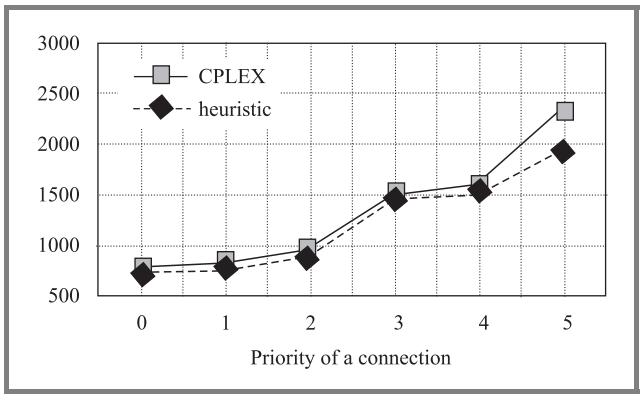


Fig. 9. Aggregate number of broken connections for 6 classes of service.

4.5. Restoration time

Figures 10 and 11 show the values of the average restoration time as the function of the connection class of service. They represent the time needed to restore a connection after

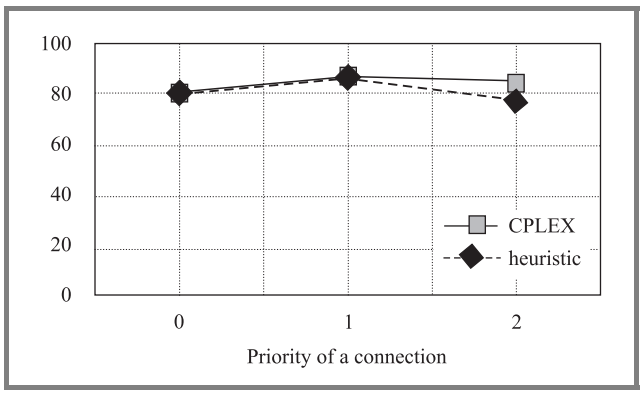


Fig. 10. Average restoration time [ms] for 3 classes of service.

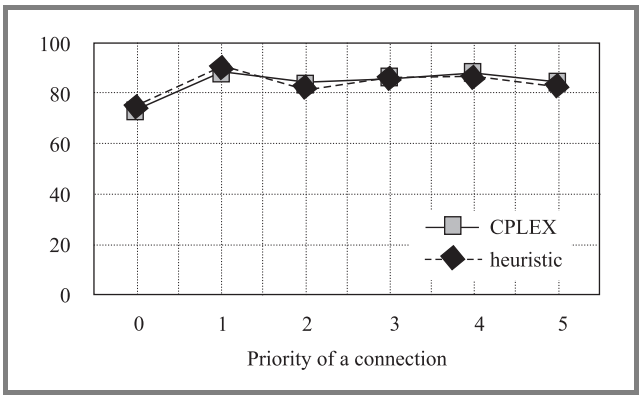


Fig. 11. Average restoration time [ms] for 6 classes of service.

a failure of a node, according to the protocol described in [11]. They prove that for the SFN network the value of restoration time does not depend much on the service class. This dependency is, however, very remarkable when analyzing the aggregate value of restoration time. Such an aggregate value for a given class k is calculated as the sum of all restoration times for connections of class k during one experiment.

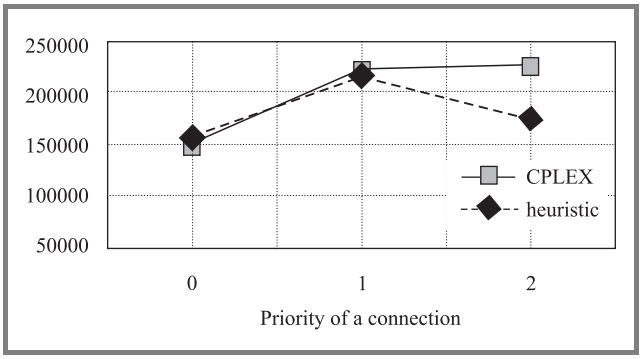


Fig. 12. Aggregate restoration time [ms] for 3 classes of service.

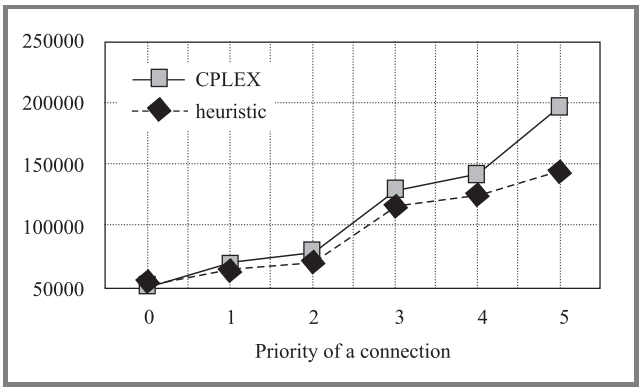


Fig. 13. Aggregate restoration time [ms] for 6 classes of service.

Figures 12 and 13 prove that the proposed model leads to a significant reduction in the value of the aggregate restoration time. The higher the requested level of service con-

tinuity is, the aggregate value of restoration time is more reduced (even about 4 times for the class 0, when 6 classes are used).

5. Conclusions

Results obtained by us confirm that the proposed model contributes to a significant reduction in the number of broken connections for classes of high priority. The active paths of such connections omit central nodes and are thus more resistant to attacks. On the other hand, low priority connections are often exposed to attacks. They are, however, realized by the shortest active paths (in the sense of distance), providing shorter values of data transmission delay.

Concluding the paper, we point out that in a scale-free network, one cannot have connections that are simultaneously resistant-to-attack and are realized by short active paths. A short active path of a connection means that with high probability it goes through central nodes and thus is at high risk of breaking.

However, we claim that establishing attack-resistant connections, realized by short active paths could be possible for networks of a regular topology. For the scale-free network it would be the best to make it more regular. Topology improvements are, however, beyond the scope of this paper and constitute the subject for future research.

References

[1] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks", *Rev. Mod. Phys.*, vol. 74, pp. 47–97, 2002.

[2] A. Barabási and R. Albert, "Emergence of scaling in random networks", *Science*, vol. 286, pp. 509–511, 1999.

[3] R. Bhandari, *Survivable Networks – Algorithms for Diverse Routing.* Boston [etc.]: Kluwer, 1999.

[4] Q. Chen and D. R. Shi, "The modeling of scale-free networks", *Phys. A*, vol. 335, *Elsev. Sci. B*, pp. 240–248, 2003.

[5] T. Gierszewski and J. Rak, "Scale-free networks", Scientific Report, Gdańsk University of Technology, 2004.

[6] K.-I. Goh, E. Oh, B. Khang, and D. Kim, "Classification of scale-free networks", *Proc. Natl. Acad. Sci.*, vol. 99, pp. 12 583–12 588, 2002.

[7] R. Kawamura, "Architectures for ATM network survivability", *IEEE Commun. Surv. Fourth Quart.*, vol. 1, no. 1, 1998.

[8] Z. Liu, Y-Ch. Lai, N. Ye, and P. Dasgupta, "Connectivity distribution and attack tolerance of general networks with both preferential and random attachments", *Phys. A*, vol. 303, *Elsev. Sci. B*., pp. 337–344, 2002.

[9] M. Pióro and D. Medhi, *Routing, Flow and Capacity Design in Communication and Computer Networks*. Amsterdam [etc.]: Morgan Kaufmann, 2004.

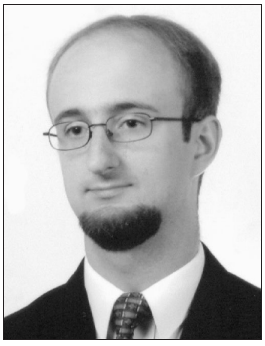
[10] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks", Part I – "Protection", in *Proc. IEEE INFOCOM*, New York, USA, 1999, pp. 744–751.

[11] S. Ramamurthy and B. Mukherjee, "Survivable WDM mesh networks", Part II – "Restoration", in *Proc. IEEE Integr. Circ. Conf. ICC*, Vancouver, Canada, 1999, pp. 2023–2030.

[12] J. W. Suurballe, "Disjoint paths in a network", *Networks*, vol. 4, pp. 125–145, 1974.

Wojciech Molisz joined the Gdańsk University of Technology, Poland, in 1968, where he is employed until now. In 1980, he was invited by the International Telecommunications Union (ITU) to the Institute of Telecommunications, Oran, Algeria, to give lectures on computer networks. From 1991 to 1993, he joined the Nuclear Research Centre, Karlsruhe, Germany, where he was involved in two international research projects in the frames of the ESPRIT programs. In the VOICE II Project (ESPRIT III), he served as the workpackage leader. Doctor of Science W. Molisz is the author and co-author of four monographs and an academic book "Solution Methods of Optimization Problems" and the author of more than 100 papers, reports and conference proceedings. His current interests are in the survivability of broadband networks, especially in the optical communications networks.

e-mail: womol@eti.pg.gda.pl
Faculty of Electronics, Telecommunications
and Informatics
Gdańsk University of Technology
Narutowicza st 11/12
80-952 Gdańsk, Poland



Jacek Rak received the M.Sc. degree in computer science from Gdańsk University of Technology (GUT), Poland, in 2003. Since 2003 he has been working as an Assistant at GUT. He is currently working toward the Ph.D. degree in computer science at GUT. His current research areas include: routing, design, dimensioning and analysis of high speed (particularly wavelength routed) backbone networks with focus on survivability.

e-mail: jrak@pg.gda.pl
Faculty of Electronics, Telecommunications
and Informatics
Gdańsk University of Technology
Narutowicza st 11/12
80-952 Gdańsk, Poland

Paper

New model of identity checking in telecommunication digital channels

Piotr Gajewski, Jerzy Łopatka, and Zbigniew Piotrowski

Abstract— We proposed an OFDM and watermarking based technology system for correspondent identity verification (CIVS) in military telecommunication digital channels. Correspondent personal identity signature (CPIS) is represented by digital watermark. The main idea of this system solution is to verify the end user who sends acoustic signal, e.g., speech, music, etc., via Internet, HF/UHF radio, modem, etc. OFDM modulation scheme is used to prepare secret digital signature. This signature is a single-use secret key used for correspondent verification, thus binary sequence of that key is changing for every session. We describe transmitter and receiver block scheme. The results of experiments for both ideal and degraded signals are described in details too. The results are summarized with comments and conclusion.

Keywords—identity checking, watermark, watermark transceiver, watermark receiver, OFDM generator, watermarked host quality, ITU-BS1116-1 test, subjective quality test.

1. Watermarking eruption

At present we are the witnesses of large watermarking application eruption. The efficiency of the personal computers enables realization of even very computational expensive algorithms. The goal of the watermarking technology is to hide additional signal under another, host signal. This additional signal can be a clear or encrypted sequence stream. Main methods for audio watermarking are well documented [1–5]. Following schemes are the most popular:

- phase modulation,
- spread spectrum,
- quantized index modulation o frequency and amplitude keying,
- echo modeling,
- least significant bit (LSB) coding.

Goals of actual investigations conducted in many laboratories concentrate on finding optimal compromise between: robustness, data payload and watermark transparency. The ideal watermarking system can produce signal transparent for human auditory system (HAS) [7], enables hidden message reliability and robustness against intentional attacks. A potential watermarking application for military purposes can be, a system for correspondent identity verification (CIVS). The main CIVS feature is message sender authorization in telecommunication channels. CIVS were designed and implemented as software in Matlab 7.0 environment and was tested in various acoustical environment conditions.

2. System for correspondent identity verification scheme

Figure 1 shows a scheme of CIVS transmitter and receiver. Digital signature (CPIS) is a unique binary sequence, dedicated for only one session. This correspondent personal identity signature (CPIS) is coded with Reed-Solomon-correspondent identity personal signature (RS-CIPS) procedure to make CPIS robust against errors at the receiver side. Six additional bits were entered to the output RS-CIPS to ensure proper detection of false positive errors.

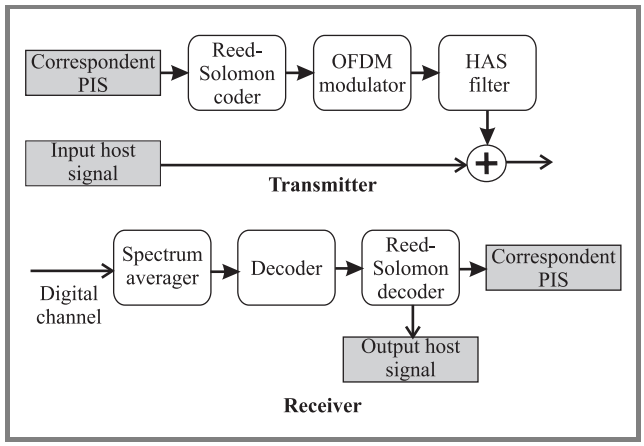


Fig. 1. CIVS transmitter and receiver scheme.

An orthogonal frequency division multiplexing (OFDM) modulator generates a signal according to RS-CIPS. Basic OFDM modulation scheme is illustrated in Fig. 2.

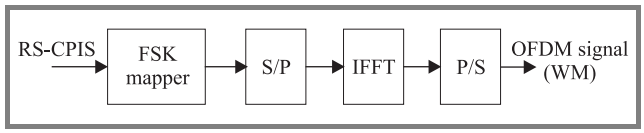


Fig. 2. Basic OFDM modulation scheme.

Binary stream in RS-CIPS form, modulate frequency shift keying (FSK) mapper, that decides which fast Fourier transform (FFT) bin should be filled. Serial to parallel conversion is required to process spectrum by IFFT function. Signal need to be converted into serial form to create an OFDM signal in time domain. Orthogonally formed harmonics are shown in Fig. 3.

Watermark signal (WM) is located in 4 kHz frequency band. Power spectrum density function (PSD) of the host signal is cumulated on this region, and it is more efficient to hide relatively stronger additional signal (WM) under host signal in this region. HAS filter contains build-in MPEG-1 audio analysis procedure to compute just noticeable difference level (JND).

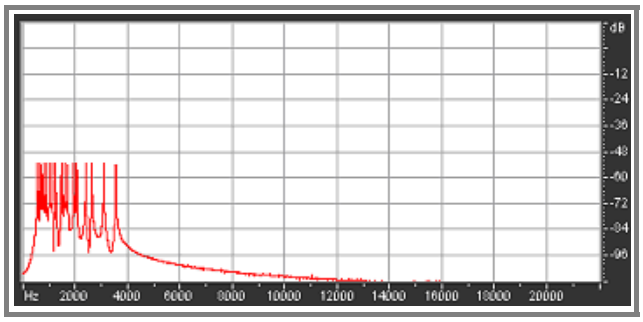


Fig. 3. OFDM generated watermark.

Signal that possesses less PSD then estimated JND is not noticeable by HAS at host signal presence. Input host signal is processed by MPEG-1 algorithm to inform HAS filter about host's JND threshold level. HAS filter is a two-stage, frequency band corrector. First stage of HAS filtering is a watermark spectrum shaping according to the host signal spectrum. The second one is correcting a power level of watermark spectrum below estimated JND threshold. Two-stage of HAS filtering is shown in Figs. 4 and 5.

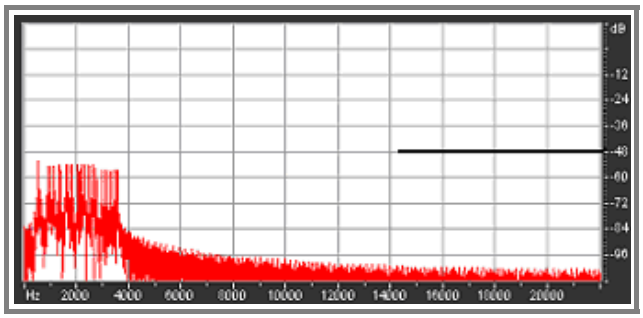


Fig. 4. First stage of HAS filtering: spectrum shaping.

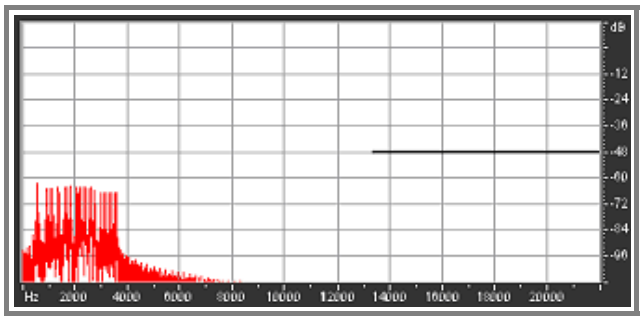


Fig. 5. Second stage of HAS filtering: level correction.

The effect of the HAS filtering can be illustrated by comparing WM before and after two-stage correction process (Figs. 6 and 7).

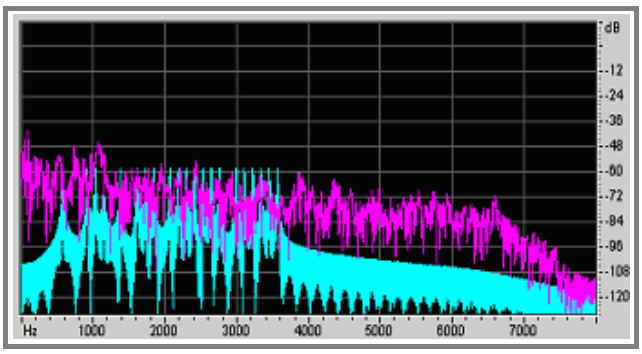


Fig. 6. WM before HAS filter correction.

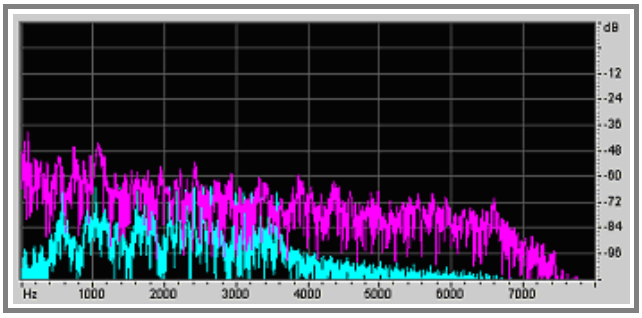


Fig. 7. WM after HAS filter correction.

Watermark receiver bases on main module: coherent spectrum averager. Spectrum averaging can reduce noise by reducing its standard variation for uncorrelated components. In this application the noise is represented by a host signal, which is not correlated contrary to WM, because WM is periodically generated according to the same pattern. Watermark gain SNR_{coh} [6] in this method is given as:

$$SNR_{coh} = \frac{\delta_{org}}{\delta_{org}/\sqrt{M}} = \sqrt{M}, \tag{1}$$

where:

δ_{org} – standard deviation of the original signal,

M – number of averaging frames.

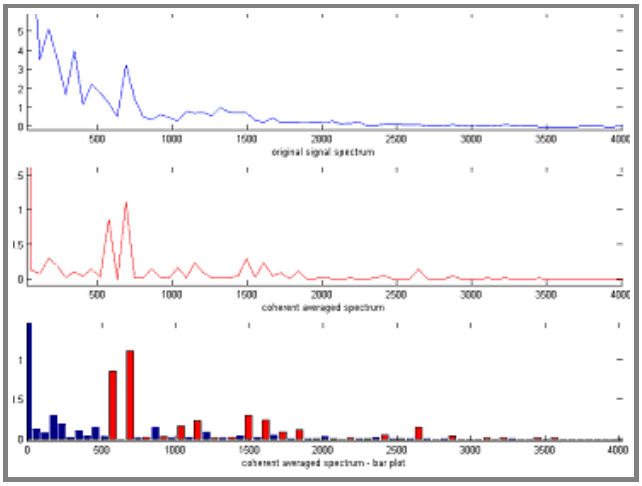


Fig. 8. Spectrum of the watermarked host, before and after coherent averaging.

We can notice, that SNR_{coh} is proportional to the square root of the M number averaging frames. Decoder, presented in Fig. 1, is responsible for correct decision. Decoder rule bases on WM level detection. The binary output stream is error corrected in Reed-Solomon decoder, thus correct CPIS should be received. The result of the coherent averaging process is shown in Fig. 8.

3. Quality of watermarked signal

Coding WM in the audio host signal we agree on degradation of this signal. The standard ITU-BS1116-1 test [8] can estimate quality of watermarked signal and degree of its degradation. In our test set of 22 listeners were asked to assess quality of tracks, watermarked by CIVS. ITU-BS1116-1 describes in details full procedure and environment conditions to be fulfilled to get reliable results. Five points degree scale is used for watermarked signal quality estimation (Table 1).

Table 1
Grading scale for ITU-BS1116-1 test

Impairment	Grade
Imperceptible	5.0
Perceptible, but not annoying	4.0
Slightly annoying	3.0
Annoying	2.0
Very annoying	1.0

One subject at a time is involved and the selection of one of three stimuli ("A","B","C") is at the discretion of this subject. The known reference is always available as stimulus "A". The hidden reference and the object are simultaneously available but are "randomly" assigned to "B" and "C" depending to the trial. Listener assesses which one from two similar signals "B" or "C" is watermarked.
Based on this grading scale, the diff grades scale (SDG) values are computed. Signal is imperceptible if SDG is higher than –1 value. In case the SGD is a positive value, it indicates that listener assessed that watermarked signal had better quality then the host. Results of carried out ITU BS1116-1 test are illustrated in Fig. 9.
We observe in Fig. 9, that group of listeners could not correctly recognize embedded watermark in tracks. None of the tested tracks achieved –1 SDG value.

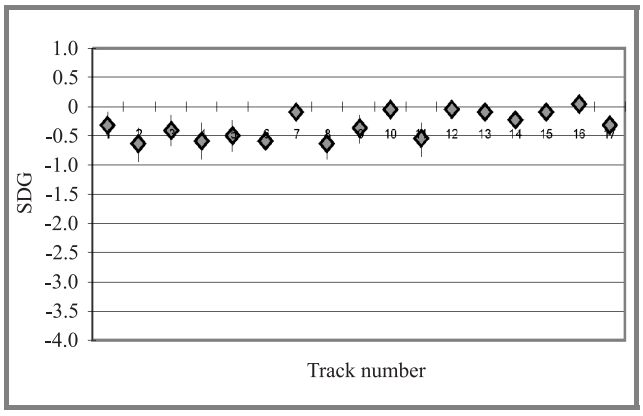


Fig. 9. SDG values for tracks coded by SCVI transmitter.

4. Test beds for CIVS

Laboratory tests were carried out for evaluation of the CPIS coding and decoding effectiveness and were conducted using HF/UHF radio stations with acoustic link on the receiver side. Internet SCVI mode based on standard voice over Internet Protocol (VoIP) is also available and passed, the same procedures carried out in radio link mode. Configurations of test beds are illustrated in Figs. 10 and 11.

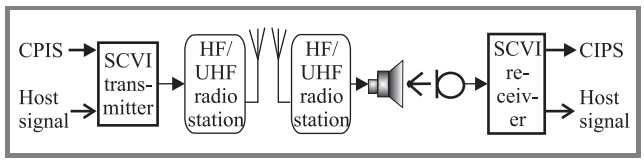


Fig. 10. Test bed CIVS – HF/UHF radio link mode.

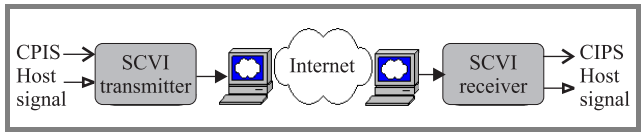


Fig. 11. Test bed CIVS – Internet VoIP mode.

The CIVS works in HF/UHF radio link with acoustic link on the receiver side, or in Internet using voice over Internet Protocol. Radio link mode with acoustic link requires very stable and precise quartz clocks for timing a/d and d/a converters with ±1 ppm short-time stability. This requirement must be fulfilled to ensure coherention in spectrum averager module.

5. Decoding host signal without embedded CPIS

One of the critical imposed requirements is SCVI robustness against wrong decision (false positive errors). When host signal is not watermarked, decoder must detect this fact and inform user that correspondent is not authorized for lack of valid CPIS. We examined decoding process in various acoustic environments (Fig. 10): silent, office, HMMV. Results of CIPS decoding (27 bits length) are illustrated in Figs. 12, 13 and 14. Figures show the performance of the CIVS decoder for various input signal duration time 1 s, 10 s, 20 s and 30 s. Thus we can determine the number of M iterations used in formula 1 and SNR_{coh}. The decision that CIPS is not embedded in a host signal is based

on a number of properly decoded "pilot" bits (p). Transmitter embedded six pilot bits and four or more (up to 6) correct decoded bits are enough to detect a fact that host contains a hidden signature. Overall number of decoded bits is indicated as (b) in Figs. 12, 13, and 14, and symbols of the three decoded tracks were indicated as 001, 004 and 019.

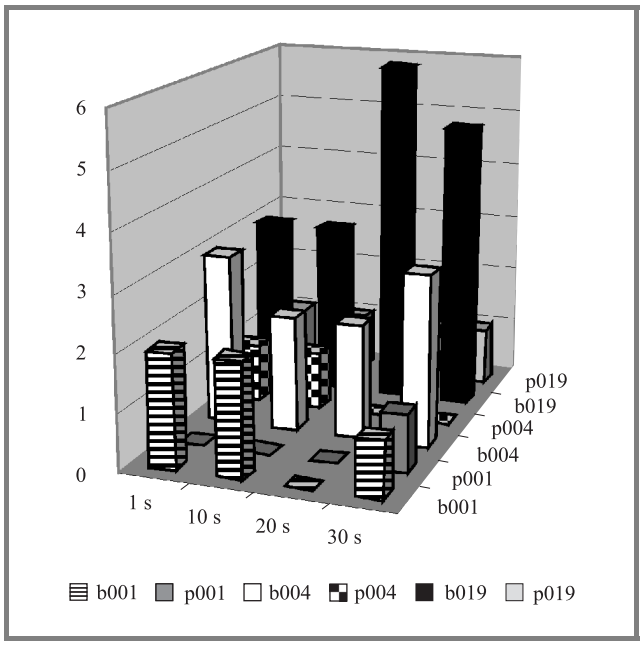


Fig. 12. Decoding host signal without embedded CIPS. Acoustic signal not degraded (silence, $SNR = 10$ dB).

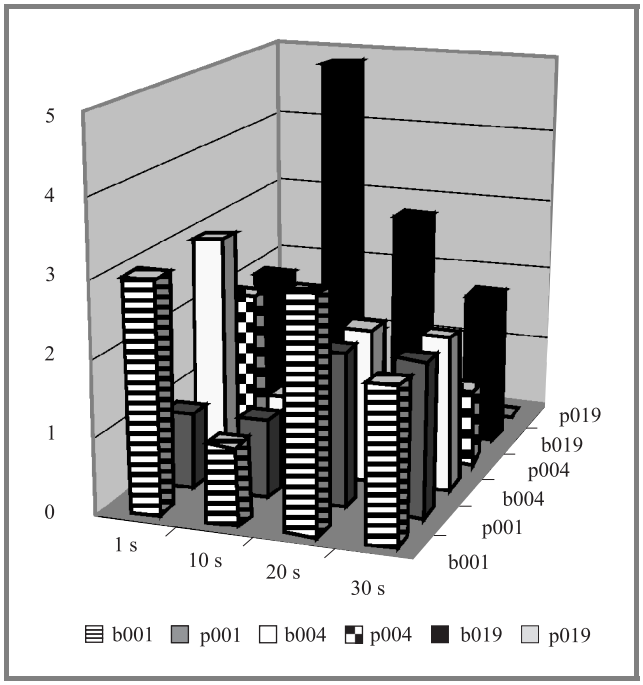


Fig. 13. Decoding host signal without embedded CIPS. Acoustic signal degraded (office, $SNR = -10$ dB).

Summarizing, CIVS will not detect CIPS in host signal until host signal will be coded by this system. Thus host signal itself can not produce false-positive decision ("silent") even if it is degraded by strong (–10 dB) degrading signal ("office", "HMMV").

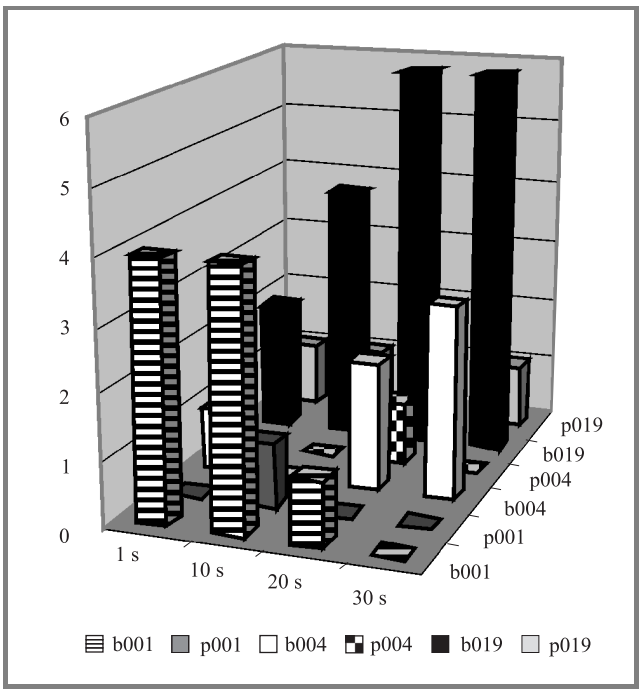


Fig. 14. Decoding host without embedded CIPS. Acoustic signal degraded (HMV, $SNR = -10$ dB).

6. Decoding watermarked host signal

We verified the correctness of CIPS decoding when host signal is watermarked by CIVS. We examined decoding process in various acoustic environments: silent (25 tracks), office (3 tracks), high-mobility multi-purpose vehicle (HMMV) (3 tracks) and for various input signal duration time: 20 s and 30 s. Results of decoding CIPS (27 bits length) are illustrated in Figs. 15, 16 and 17. Decoder gives correct CIPS even for very strong degraded signal (–10 dB). In all cases CIVS detects a fact that sig-

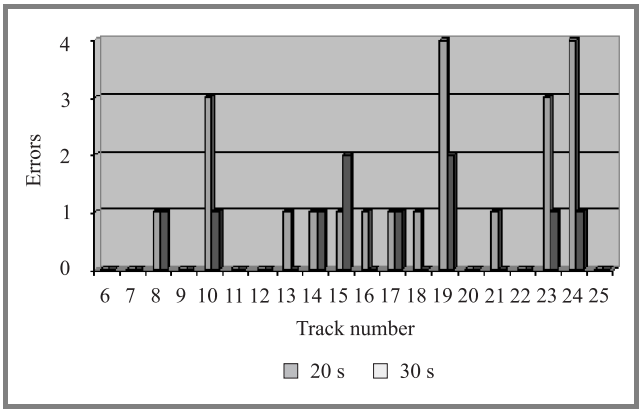


Fig. 15. Decoding host signal with embedded CIPS. Acoustic signal not degraded (silence, $SNR = 10$ dB).

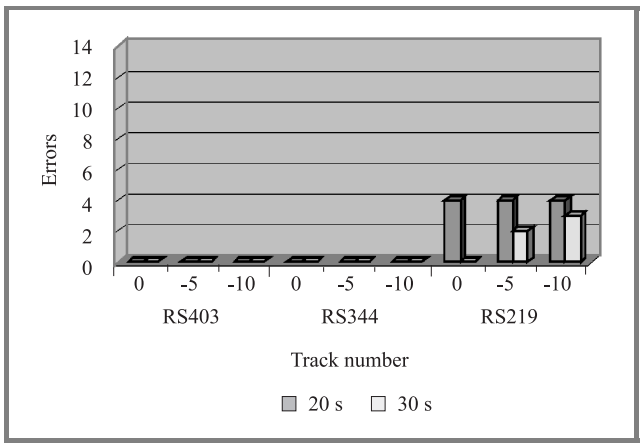


Fig. 16. Decoding host signal with embedded CIPS. Acoustic signal degraded (office).

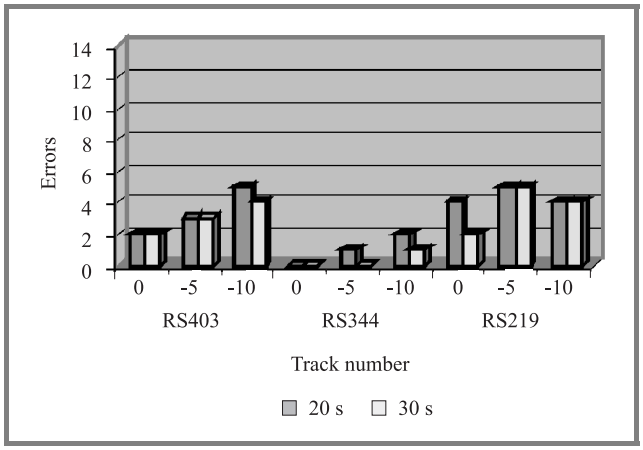


Fig. 17. Decoding host signal with embedded CIPS. Acoustic signal degraded (HMMV).

nal is watermarked. Because proposed Reed-Solomon code scheme has ability to correct only 3 bits, the figures where number of errors is higher than 3, decoded CIPS was not correct.

7. Just noticeable difference level changing

We verified the correctness of CIPS decoding when host signal is watermarked by CIVS using various modifications of JND level. Signal-to-mask ratio (SMR) coefficient describes proportion between power of host signal and WM. Standard MPEG-1 psychoacoustic procedure compute SMR = JND level for audio host signal and this is optimal value from listener point of view, but when SMR level is smaller we can decode MW with greater efficiency (higher WM power) and with higher probability that HAS recognize watermarked signal. In opposite, higher SMR can reduce watermark HAS detection probability to zero but correct WM detection will be impossible because of too small WM power. In this experiment we correct JND level using values: –9 dB –6 dB –3 dB 0 dB 3 dB 6 dB and 9 dB thus 0 dB equals JND level computed for MPEG-1. Duration time 10 s for each from three tested tracks was assumed. Results of this experiment are illustrated in Figs. 18, 19, 20 and 21. We can notice that

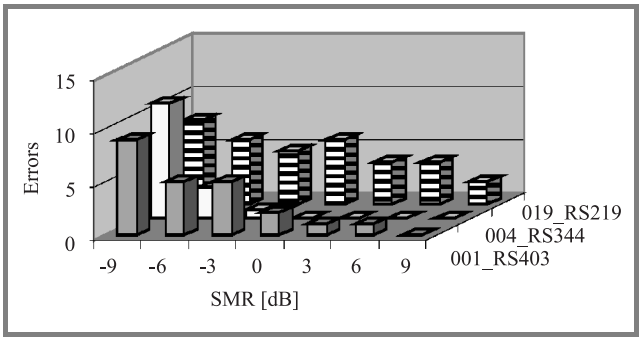


Fig. 18. Errors in SMR function.

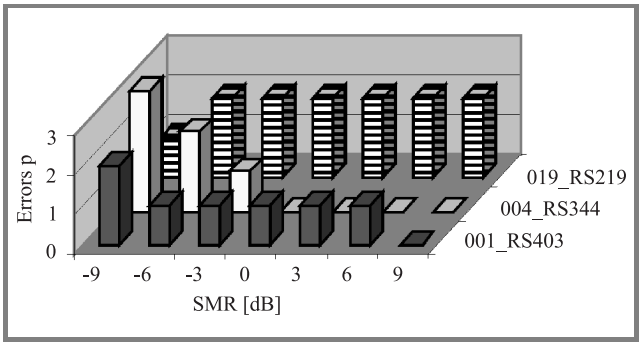


Fig. 19. Pilot errors in SMR function.

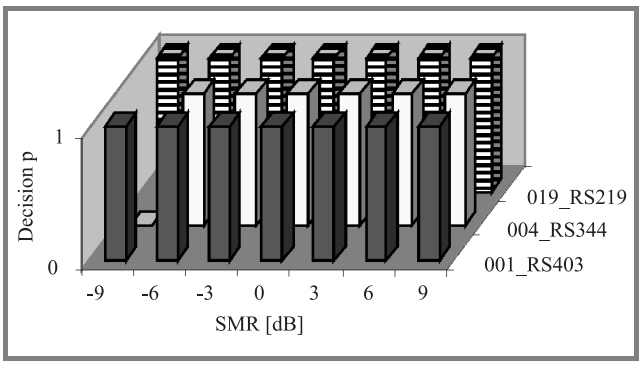


Fig. 20. CIVS decisions for WM presence detection.

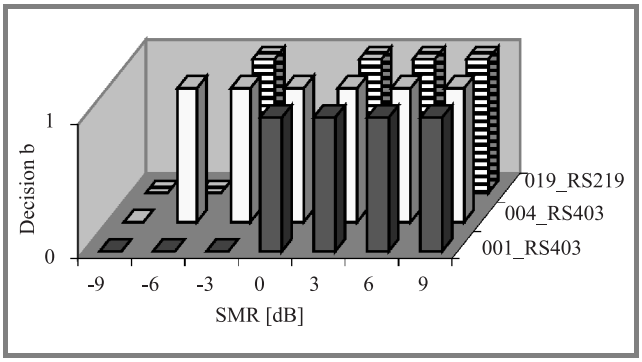


Fig. 21. CIVS decisions for correct CPIS decoding.

CIVS is very robust for false-negative detection. SMR below JND level (0 dB) reduce effectiveness (decoding for 10 s averaging).

8. Conclusion

Presented system proves its usability at the laboratory tests. The correspondent ID signature can be detected with success (watermark transmitted in digital channels, even strongly degraded). Future experiments will concentrate on higher system effectiveness and robustness for loose compression and higher number of embedded bits. CIVS can be used in those military systems where identification has a priority before message interpretation. Implementing hash function for host signal we can get a message authentication system (MAS) and together with CIVS mechanism it enables, high priority message will never be anonymous and always will be integral.

References

[1] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*. San Francisco: Academic Press, 2002.

[2] M. Arnold, "Audio watermarking: features, applications and algorithms", in *IEEE Proc. 2000*, Darmstadt: Department for Security Technology for Graphics and Communication Systems Fraunhofer-Institute for Computer Graphics, 2000.

[3] W. Bender, D. Gruhl, N. Morirrnoto, and A. Lu, "Techniques for data hiding", *IBM Syst. J.*, vol. 35, no. 3-4, pp. 313–336, 1996.

[4] C. Xu, J. Wu, and Q. Sun, "Digital audio watermarking and its applications in multimedia database", in *Proc. ISSPA'99*, Brisbane, Australia, 1999.

[5] F. A. Everest, *Master Handbook of Acoustics*. New York: McGraw-Hill, 2001.

[6] R. G. Lyons, *Understanding Digital Signal Processing*. New York: Addison Wesley Longman, 1997.

[7] E. Zwicker, *Psychoacoustics*. New York: Springer-Verlag, 1982.

[8] *Methods for the subjective assessment of small impairments in audio systems including multichannel sound systems*, ITU-R BS.1116-1.

Piotr Z. Gajewski received the M.Sc. and D.Sc. degrees from Military University of Technology (MUT) Warsaw, Poland, in 1970 and 2001, respectively, both in telecommunication engineering. Since 1970 he has been working at Electronic Faculty of Military University of Technology (EF MUT) as a scientist and lecturer in communications systems (radios, cellular, microcellular), signal processing, adaptive techniques in communication and communications and information systems interoperability. He was an Associate Professor at Telecommunication System Institute of EF MUT from 1980 to 1990. From 1990 to 1993 he was Deputy Dean of EF MUT. Currently he is the Director of Telecommunication Institute of EF MUT. He is an author (co-author) of over 80 journal publications and conference papers as well as four monographs. He is a member of the IEEE Vehicular Technology and Communications Societies. He is also a founder member of the Polish Chapter of Armed Forces Communications and Electronics Association.
e-mail: pgajewski@wel.wat.edu.pl
Military University of Technology
S. Kaliskiego st 2
00-908 Warsaw, Poland

Jerzy Łopatka received the M.Sc. and Ph.D. degrees in communications from the Military University of Technology (MUT), Warsaw, Poland. At present he is a Head of Radiocommunication Section in the Telecommunication Institute (MUT). His main research interests include digital signal processing in wireless systems.
e-mail: jlopatka@wel.wat.edu.pl
Military University of Technology
S. Kaliskiego st 2
00-908 Warsaw, Poland

Zbigniew Piotrowski received the M.Sc. and Ph.D. degrees in communications from the Military University of Technology (MUT), Warsaw, Poland, in 1996 and 2005, respectively. At present he is a DSP engineer of Radiocommunication Section in the Telecommunication Institute (MUT). His main areas of interest are speech and audio processing, telecommunication systems engineering and watermarking technology.
e-mail: zpiotrowski@wel.wat.edu.pl
Military University of Technology
S. Kaliskiego st 2
00-908 Warsaw, Poland

Paper

Planning the introduction of IPv6 in NATO

Robert Goode

Abstract— The NATO wide area network provides secure IP services to NATO commands and agencies, and offers information exchange gateways to nations and coalition operations. The IP services support the NATO-wide deployment of core automated information systems (AIS), and the placement of specific functional area services (e.g., intelligence, logistics, C2IS for the services, etc.) at commands. To maintain and improve interoperability within NATO and with partners, NATO will transition from version four of the Internet Protocol (IPv4) to version six (IPv6). The transition to IPv6 will involve the IP network, the information exchange gateways, the core AIS, the functional area services, and the supporting CIS infrastructure. The IPv6 naming and addressing plan being developed supports the NATO command structure and interoperability with NATO partners. The critical issue in the planning process is to support the incremental introduction of IPv6 whilst maintaining network security and reliable interworking with existing IPv4 systems and limiting increases in operations and maintenance costs. To minimise costs and maximise effectiveness NATO is planning the transition in a timescale that is commensurate with commercial adoption in NATO countries, the technology refreshment points for major systems, and the availability of IPv6 security components. New NATO projects will prepare for the transition by detailing their IPv6 upgrade path and procuring dual stack (IPv4 and IPv6) equipment. NATO will develop and adopt standardised approaches for IPv6 protocols and network design.

Keywords— Internet Protocol, IPv6, transition, NATO, CIS.

1. Introduction

The NATO operates a broad range of communications and information systems (CIS) at NATO headquarters (HQ), organizations and agencies. The sites are linked by the NATO secret wide area network (NSWAN), which provides a NATO-wide, cost-effective, interoperable and secure capability. NATO also operates the NATO unclassified WAN (NUWAN) and a number of mission/theatre classified WANs (MSWAN). The NATO WANs provide cryptographically protected virtual private networks (VPN). The traffic on the plaintext (high side) side of the encryption device is referred to as "red", whilst the enciphered traffic (on the low side) is referred to as "black". The terms "red" and "black" are used in this paper to refer to these two cryptographically separated routing domains as shown in Fig. 1.

The NATO CIS are divided into core area services (CAS), which are used by all NATO users, and the functional area services (FAS), which are role-based applications. The CAS provides NATO-wide automated information applications such as electronic mail, web services and document preparation tools. The FAS support specific functions such as logistics, ground, maritime and air operations, intelligence services, etc. The NATO CIS interfaces to national fixed and mobile networks to cover the whole NATO area to support high level political consultation and command and control of military forces. NATO CIS is being transformed to achieve the NATO network enabled capability (NNEC) with a seamless flow of information, and to support the NATO response force (NRF). The NATO response force will be a coherent, high readiness, joint, multinational force package, technologically advanced, flexible, deployable, interoperable and sustainable. As part of the ongoing CIS transformation NATO is planning for a transition of the packet switched NATO VPN (NVPN) from version four of the Internet Protocol (IPv4) to version six (IPv6) [1].

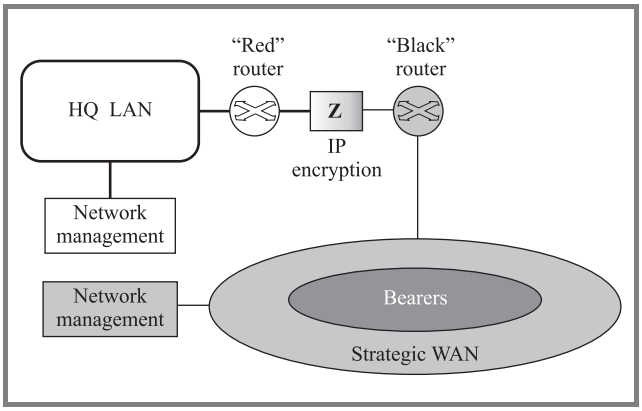


Fig. 1. "Red" and "black" routing domains.

At the time of writing three NATO nations (FR, GE, US [2]) have issued directives relating to the use of IPv6 in their national defence infrastructure, and the US has directed the use of IPv6 in other government departments [3, 4]. The Commission of the European Communities issued a communication to the Council and the European parliament in 2002 [5] which called upon member states to encourage transition towards IPv6. All major vendors of network routers support IPv6, and the vendor of the dominant operating system for PCs (Microsoft) has stated that the next major update to the Windows platform, due out in 2006, will use IPv6 as the preferred transport [6].

The main goals for the NATO IPv6 transition are to:

- support the NNEC seamless flow of information;
- maintain and improve interoperability;

– take advantage of new capabilities to increase functionality and reduce cost;

– stay in line with commercial developments.

This paper considers the planning necessary to achieve the NATO IPv6 transition goals, which involves a pervasive change across the whole of NATO CIS and interfaces to NATO nations and NATO partners. The aims and objectives of the transition planning are presented, and an outline is given of the technical areas being considered. This paper focuses on the transition planning for the communications systems, and discusses planning guidance to NATO and national users at the strategic and tactical level on using the IPv6 NATO WANs.

2. The IPv6 transition planning – outline

The NATO IPv6 transition planning will:

- Develop an evolutionary IPv6 transition plan for NATO CIS infrastructure:
  - specify the IPv6-support to be built into the NATO WANs;
  - specify the approach for naming, addressing, routing, network management, security and transition mechanisms in the NATO WANs.
- Determine the manner in which interoperability will be maintained with NATO partners during the transition:
  - develop guidance to NATO partners on interworking with NATO during the transition.
- Provide NATO with the concepts and know-how to migrate the CIS across strategic and deployed systems to work on a single virtual IPv6 network:
  - develop guidance to core and functional area services to become IPv6-ready;
  - identify the standards which must be supported in specific functional elements.
- Identify new capabilities in IPv6 of which NATO can take advantage:
  - examine: multicast, anycast, multiple address plans, radically increased address space, auto configuration, mobility support, flow labelling, etc.
- Determine the timelines and approaches which achieve the best cost-benefit for the transition in a timescale commensurate with the commercial adoption in NATO countries:
  - work with NATO nations, partners, and industry on timeline planning.
- The transition planning is broad in scope to introduce the system, technical and operational views which need to be considered due to the pervasive nature of an IP transition. In order to support the broad nature of the planning process, the follow methods are used:
  - technical studies;
  - NATO working groups with representation from all NATO stakeholders;
  - in-house test-beds and multinational experimentation;
  - participation in IPv6-related forums and events;
  - IPv6-related training.

3. IPv6-support in the NATO WANs

The NATO WANs must maintain full support for existing IPv4 services during the transition period, to avoid breaks in operational service. This means that the IPv6-support must be in parallel to the IPv4 support, and must not negatively impact it. A second requirement is that the IPv6 access must be ubiquitous, rather than constrained to specific network access points. The transition to IPv6 is envisaged as evolutionary, with an initial low level of IPv6 traffic, which increases over the lifetime of the transition. The transition period is expected to be measured in decades because of the need to maintain IPv4 support to inter-work with legacy systems. The IPv6 support must thus scale from minimal usage to being the dominant traffic type, and should do so in a manner that is cost-effective over the lifetime of the transition.
The NATO WANs need to support routing of IPv6 traffic in an efficient manner, and name resolution through an IPv6-enabled domain name service (DNS), which needs to operate effectively in parallel to an IPv4-enabled DNS. The whole network must be operated securely with guard technology to protect against external network attacks, and intrusion detection to monitor the internal integrity of the environment.

3.1. Naming structure

The fully qualified domain names applied to network devices are frequently visible to users (for example in uniform resource locators – URLs) and so need to make sense to non-technical staff, as well as supporting the needs of network managers. The naming structure is often driven by organizational structure, and uses a standardized format for naming devices types (routers, switches, workstations) and usage (mail server, firewall, administrator, etc.).
The transition from IPv4 to IPv6 does not intrinsically alter the organizational structure or application usage; therefore the existing IPv4 naming structure will be applied to IPv6. The approach clearly simplifies the network manager's task of identifying a specific device in both the IPv4 and IPv6 network. The approach also means that

the user does not need to know whether an application is being accessed via IPv4 or IPv6, as the same name can be used in both cases. De-conflicting the resolving of DNS queries which may result in an IPv4 or an IPv6 address (or both) places some constraints on the deployment of the IPv6.

3.2. Addressing plan

Numerical representations of IPv4 addresses are usually hidden from end users, who use the human-readable names instead. The addressing plan can thus be divorced from organizational structure and the use to which a network element is put; and be driven by the network structure to improve operating efficiency and easy maintenance. Two significant considerations for the addressing plan are aggregation to reduce routing table size and frequency of routing advertisements, and scalability to support growth (both planned and exceptional). An addressing plan therefore tends to be hierarchically constructed along geographic (or connectivity) lines, and have reservations for future growth. The transition from IPv4 to IPv6 does not intrinsically alter the network structure or growth forecast, therefore the existing IPv4 addressing plan format will be applied to IPv6. This may mean that a simple mapping function can be used to map hierarchical elements of the IPv4 address onto equivalent elements of the IPv6 address. Clearly there are differences between the IPv4 and IPv6 address formats defined by the Internet Engineering Task Force (IETF), such as the number of bits and the manner in which the addressing mode (unicast, multicast, globally routable versus private/link-local, etc.) is encoded in the bits, and these must be taken into account. Some new capabilities in IPv6 which must be assessed are the option to have multiple addresses plans (with multiple addresses per network interface), the use of anycast, and renumbering.

3.3. Routing

The NATO WANs require an interior gateway protocol (IGP) for distributing routing information internally. An exterior gateway protocol is required for exchanging routing information with peer networks. NATO currently has a limited requirement for IP multicast, which seems likely to increase to achieve the NNEC vision of seamless information exchange. The "red" routing domain is separated from the "black" routing domain by IP-based encryption devices, but both routing domains must be co-operatively managed to achieve a stable and robust network that can support the required network quality of service.

3.4. Network management

A critical element of a reliable CIS infrastructure is the network management system. The network manager needs to view the traffic load and health of the distributed network elements in order to perform problem identification and resolution, and to plan provisioning schedules. The network management system will need to be dual-stacked to provide the monitor and control interface for both the IPv4 and the IPv6 components. An approach is required that will achieve harmonized network management of both the "red" and "black" domains for the NATO WANs. A sample of the requirements is given below:

- Automated address space management for both IPv4 and IPv6.
- Network monitoring and visualisation for both IPv4 and IPv6.
- Scaleable element management.
- Extensible for QoS, transition mechanisms, gateways, applications.
- Manage multiple inter-dependent networks:
  - IPv4 and IPv6 networks,
  - Enciphered virtual private networks ("red") over range of bearers ("black").

3.5. Security

Security is a strong requirement for NATO classified systems. In addition to the confidentiality requirements which can be met by a high-grade IPv4 and IPv6-capable encryption device, there are requirements for integrity, authentication, non-repudiation, reliability, auditing, intrusion detection, and physical security. The full range of high-grade security devices must be available in IPv6-capable form to work in concert with the existing IPv4 devices without significantly increasing the total cost of ownership of the secure networks.

3.6. Transition mechanisms

The IETF has issued a number of request for comment (RFC) documents, e.g. [7–9], which describe a range of transition mechanisms that meet the identified requirements for IPv6 support in the NATO WANs. The simplest initial approach is to transport IPv6 packets encapsulated in IPv4 packets (tunnelling) over the existing IPv4 infrastructure. This works well when the IPv6 traffic is sparse, as was demonstrated by the success of the 6bone [10, 11], but as an approach its suitability is inversely proportional to the quantity of IPv6 traffic. Given that the IPv6 traffic will eventually be dominant, a more suitable approach is to support native IPv4 and IPv6 traffic simultaneously by using dual-stack network elements. Consideration must also be given to converting the network cores from IPv4 to IPv6, and tunnelling IPv4 over IPv6 in the core. The cost-benefit analysis is a significant part of determining the best approach.

There are also requirements to gateway traffic between IPv4 and IPv6 systems, i.e., to interconnect them rather than just enable them to operate in parallel. The IETF has documented a number of application-level transition mechanisms [7].

There is a body of work on the advantages and disadvantages of each transition mechanism in specific circumstances, which includes guidance on transition planning [12–15]. This experience will form a valuable input to NATO.

4. New capabilities in IPv6

The design of IPv6 has benefited from decades of experience with IPv4 networks. The most visible change is that the address space has been drastically increased, from 32-bits to 128-bits. Other improvements have also been made, in the areas of multicast, anycast, mobility, and autoconfiguration. There is also a field which traffic sources can use to label flows through the network which may offer practical benefits to NATO networks by enabling a richer support for network quality of service than is possible for IPv4 (see for example [16] for the issues, [17] and [18] for a possible way forward).

5. Guidance to information services

The purpose of enabling IPv6 support in the NATO WANs is to facilitate IPv6 applications. One obvious part of the guidance to the information service developers is to port their networked applications to an IPv6 stack. It is additionally necessary to provide guidance on inter-working IPv4 with IPv6, including information on when to use specific approaches out of the range on transition mechanisms available. One example of such guidance is [15].

6. Guidance to NATO partners

Maintaining and improving interoperability with NATO nations and partners is a key driver for the transition to IPv6 by NATO. The exchange of information between NATO and a nation or organization is achieved through information exchange gateways (IEGs) [19, 20] as shown in Fig. 2.

The IEGs implement application-level proxies and guard functions for web and electronic-mail, thereby enabling controlled release of data. Applications which require additional services through the IEG can develop the necessary application-level proxies and accredited guard functions. The IEGs do not provide a general packet routing service, but instead form an IP break. This means that routing information does not flow between NATO and nations or partners through an IEG.

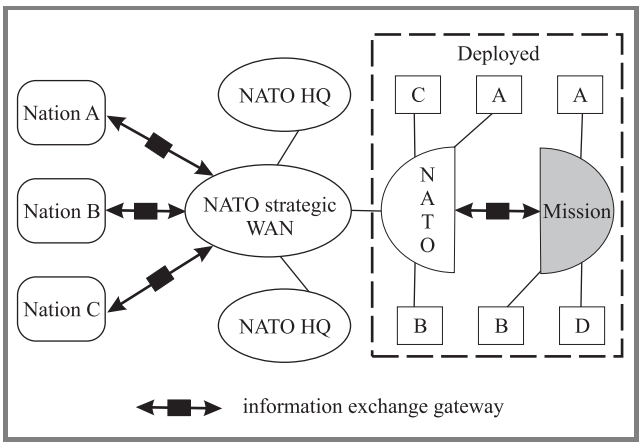


Fig. 2. Use of information exchange gateways.

7. The IPv6 compliance

A common definition of IPv6 compliant that can be uniformly applied in procurement of NATO common-funded equipment is a pre-requisite to achieving a fully functional IPv6 network, and a vital part of defining a standardized interconnection point for NATO and national systems. Work on this topic has been performed in a number of fora, including the IPv6 ready program [21], the European Community [22], and the US DoD [23]. NATO will build on this body of work to develop a common NATO definition in consultation with the nations.

8. Experimentation

NATO actively utilizes testing, experimentation and exercises to support interoperability testing of NATO and national systems. Relevant activities include the NATO interoperability environment testing infrastructure (NIETI), the annual coalition warrior interoperability demonstration, combined endeavour, the *Interoperable Networks for Secure Communications* (INSC) project, and the Combined Federated Battle Laboratories Network (CFBLNet). INSC [24] is an eight-nation project to develop the future communications architecture for combined joint out-of-area operations, and it has an IPv6-focus [25]. The CFBLNet [26] is an arrangement between the US, Combined Communications-Electronics Board (CCEB) and NATO to provide the network of choice for test and evaluation experimentation. The charter nations/organisations are the US, the CCEB nations (AUS, CAN, NZ, UK, US), the NATO nations, and NATO as an organisation. The CFBLNet is currently running a multinational IPv6 initiative.

In order to achieve increased interoperability experimentation will be used to validate the operation of selected transition mechanisms, naming and addressing plans, security devices, routing approaches, etc. Such experimentation is already underway and will need to be continued for the duration of the transition, which is likely to continue for many years.

9. Training

Training of the network designers, network operators, security experts, and application developers will be required to achieve a successful transition. This will ensure that the appropriate transition mechanisms are applied in each case, and with the necessary security configurations.

10. Conclusion

This paper has introduced the areas which must be covered by the NATO IPv6 transition planning process in order to successfully manage the introduction and migration to IPv6 whilst maintaining the interoperability with existing IPv4 systems over a prolonged transition period.

References

[1] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) specification", RFC 2460, Dec. 1998

[2] J. Stenbit, "Memorandum Establishing DoD Policy for Transition to Internet Protocol Version 6 (IPv6)", US Department of Defense, ASD NII-DoD CIO, 9 June 2003.

[3] "IPv6: Federal Agencies need to plan for transition and manage security risks", Rep. GAO-05-471, http://www.gao.gov/new.items/d05471.pdf

[4] "Transition planning for Internet Protocol Version 6 (IPv6)", in Executive Office of the President, Office of Management and Budget as OMB M-05-22 Memorandum for the Chief Information Officers number M-05-22, on Aug., 2005, http://www.whitehouse.gov/omb/memoranda/fy2005/m05-22.pdf

[5] "Next generation Internet – priorities for action in migrating to the new Internet Protocol IPv6", in COM(2002) 96 final by the Commission of the European Communities on 21.02.2002 as a communication from the Commission to the Council and the European parliament, http://europa.eu.int/eur-lex/en/com/cnc/2002/com2002_0096en01.pdf

[6] L. Huang, "Microsoft IPv6 Update", in *North Amer. IPv6 Summit*, 2005, http://usipv6.unixprogram.com/North_American_IPv6_Summit_2004/052005/tue/Leigh_Huang.pdf

[7] R. Gilligan and E. Nordmark, "Transition mechanisms for IPv6 hosts and routers", RFC 2893, Aug. 2000.

[8] D. Haskin and R. Callon, "Routing aspects of IPv6 transition", RFC 2185, Sept. 1997.

[9] B. Carpenter and K. Moore, "Connection of IPv6 domains via IPv4 clouds without explicit tunnels", RFC 3056 , Febr. 2001.

[10] 6bone, http://www.6bone.net/

[11] I. Guardini, P. Fasano, and G. Girardi, "IPv6 operational experience within the 6bone", http://carmen.cselt.it/papers/inet2000/index.htm

[12] J. Dočkal and T. Fiala, "Research of the migration from IPv4 to IPv6 in the Czech Army", in *Proc. 6th NATO Reg. Conf. Milit. Commun. Inform. Syst.*, Zegrze, Poland, 2004, pp. 357–362.

[13] T. Chown, "IPv6 campus transition experiences", in *Proc. Int. Symp. Appl. Internet SAINT 2005*, Trento, Italy, 2005.

[14] M. Brig, "Integration techniques – a technical brief on the methods of transitioning to IPv6", in *US IPv6 Summit*, Reston, USA, 2004, http://usipv6.unixprogram.com/usipv6_reston_2004/tue/Brig.pdf

[15] V. Pecus, "DoD IPv6 applications transition planning guidelines", in *US IPv6 Summit*, Reston, USA, 2004, http://usipv6.unixprogram.com/usipv6_reston_2004/thu/Pecus.pdf

[16] R. Goode, P. Guivarch, and M. Stell, "Quality of service in an IP crypto partitioned network", in *Proc. MILCOM 2002*, Anaheim, USA, 2002, vol. 2, pp. 1154–1159.

[17] P. Sevenich and C. Reichmann, "Multiplexing time-critical and conventional data over tactical IPv6 networks of low bandwidth", in *INSC Symp*., The Hague, The Netherlands, 2003, http://insc.nodeca.mil.no/ifs/files/public/Symposium/Symposium

[18] M. Amanowicz, P. Sevenich, J. Jarmakiewicz, and M. Pilz, "Quality of service support in IPv6-based military networks with limited bandwidth", in *Proc. RTO IST-054 Symp. Milit. Commun.*, Rome, Italy, 2005.

[19] M. Diepstraten and R. Parker, "NATO AIS cooperative zone technologies", in *Proc. 4th NATO Reg. Conf. Milit. Commun. Inform. Syst. RCMCIS*, Zegrze, Poland, 2002, pp. 207–216.

[20] S. Cresdee, M. Diepstraten, E. Frambach, W. Hoogeveen, F. Nolden, L. Schenkels, and D. Stanley, "NATO AIS information exchange gateway evolution", in *Proc. 5th NATO Reg. Conf. Milit. Commun. Inform. Syst. RCMCIS*, Zegrze, Poland, 2003.

[21] "IPv6 ready at URL", http://www.ipv6ready.org/frames.html

[22] "IPv6 standardisation report", IST-2001-34056, M. Ford, Ed., 2005, in European Commission as deliverable D5.1.10 under WP5 as part of the EC Information Society Technologies 6LINK programme; PDF version available via the homepage of the European Commission IST IPv6 cluster, http://www.ist-ipv6.org

[23] D. Coe and A. Sekelsky, "IPv6 capable – DoD definition", in *US IPv6 Summit*, Reston, USA, 2004, http://usipv6.unixprogram.com/usipv6_reston_2004/thu/Coe-Sekelsky.pdf

[24] "Interoperable networks for secure communications (INSC)", http://insc.nodeca.mil.no/

[25] S. Gee, "Internetworking for coalition interoperability", in *9th Int. Comm. Contr. Res. Technol. Symp.*, Copenhagen, Denmark, http://www.dodccrp.org/events/2004/ICCRTS_Denmark/CD/papers/013.pdf

[26] "Combined Federated Battle Laboratories Network (CFBLNet)", http://cfbl.nc3a.nato.int

Robert Goode has been working in the area of network communications for defence systems for twenty years, split between commercial and NATO positions. He has worked on a variety of technology areas including X.25, X.400, trusted computer base, mobility, IP quality of service, and IPv6. He is a Principal Scientist at the NATO Consultation, Command and Control Agency (NC3A) where he is leading the team drafting the NATO IPv6 transition plan. He is actively involved in multinational activities examining IPv6 such as the INSC project and CFBLNet, for which he is the NATO "national" lead.
e-mail: Rob.Goode@nc3a.nato.int
NATO C3 Agency (NC3A)
Oude Waalsdorperweg 61
2597 AK The Hague, The Netherlands

Paper

Simple admission control procedure for QoS packet switched military networks

Damian Duda and Wojciech Burakowski

Abstract— Providing quality of service (QoS) into the networks based on the packet switched technologies, as ATM and IP, is currently the challenge for the military communication system designers. The main element for achieving QoS capabilities is to implement effective admission control (AC) function, which regulates the volume of submitted traffic to the network. The traditional approach for the AC is that it is invoked by each call requesting QoS. As a consequence, the call set-up latency is increasing and, in addition, the signaling traffic in the network is growing. This paper proposes a simple AC method that is based on the online traffic load measurements and assumes that the AC is involved only when the load exceeds a predefined threshold. As a consequence, for most of the connections the AC is not necessary to be executed and this causes lower set-up phase duration and limits the volume of signaling traffic. The numerical results showing effectiveness of the approach are included and compared with traditional AC performing.

Keywords— QoS, admission control, measurement based AC.

1. Introduction

The evolution scenario of military networks assumes that the packet switched networks, based on ATM/IP technologies, will substitute the existing circuits switched networks. Notice, that in the most of the NATO countries this process was finished or is currently running. Let us remark that packet networks designed for the military should provide quality of service (QoS) capabilities for the packet transmission level since some information needs to be transmitted as urgent or very urgent. So, we can not rather use the best effort service only as it is available in today's public Internet. The QoS requirements coming from the military applications are mainly referring to the timely context delivery and real time transmission capabilities for such applications as voice over Internet Protocol (VoIP), videoconference, radar data, etc.

One can distinguish between two methods for providing QoS capabilities into the packet switched networks. The more complex approach is to implement in the network additional QoS mechanisms working on different time scale, like traffic control mechanisms at the packet level (classifiers, schedulers, markers, etc.), traffic control mechanism at the call level – admission control (AC) as well as adequate QoS routing procedures. It is worth to mention that these mechanisms could be collected in the form of appropriate QoS architecture as differentiated services (DiffServ) architecture [1]. The second approach is to keep the network in the over-provisioned state. Anyway, such approach that can be effective in the public core IP-based networks is not appropriate to be applied in the military, especially on the tactical level, where we face with the radio or radio-relay links with limited capacity. As a consequence, the more attractive solution for providing QoS in the military IP-based networks is to follow the approach with engaging traffic control mechanisms, where the AC function plays a key role.

The traditional approach for the AC is that it is invoked by each call requesting QoS. As a consequence, the call set-up latency is increasing and, in addition, the signaling traffic in the network is growing. In order to overcome this problem, in this paper we propose and evaluate a new strategy for performing in an effective way AC function, targeted to limit invocation of the AC but keeping QoS capabilities. The discussed approach, named as AC with threshold (AC-T), is based on the online traffic load measurements. It assumes that the AC is invoked only when the traffic load exceeds a predefined threshold.

2. Proposed AC-T method

The strategy based on the invocation of AC each time when new call is submitted to the system is the most popular solution for achieving QoS guarantees in the network. Let us recall that the AC makes the decision about admission/rejection of new call on the basis of the traffic declarations and the current traffic load. Anyway, when decision is to accept, the network guarantees that the packet transfer characteristics will satisfy the QoS objectives as assumed for the AC. This strategy is typical for QoS networks.

The call request includes QoS needs coming from the user application, e.g., requested bandwidth, accepted packet losses, etc. The appropriate network control entities hold current status of the network resources and give the necessary information for making AC decision. The available resources are computed accordingly to the predefined QoS objectives. For example, QoS objectives for voice traffic are target values of maximum delay, maximum delay variation and packet loss ratio.

In the simplest case, when the provisioning of resources is based using static bandwidth allocation, the call request is accepted if the sum of currently used bandwidth and the bandwidth needed for new connection does not exceed

the bandwidth allocated for QoS traffic. For instance, the AC algorithm may work accordingly to the formula:

$$Bw_{new} + \sum_{i=1}^{N} Bw_i \leq C, \tag{1}$$

where:

N – number of running connection,

Bw_{new} – bandwidth requested for a new connection,

Bw_j – bandwidth used by ith connection, $i = 1, \ldots, N$.

The acceptance decision is made only when formula (1) is satisfied. It should be noticed that this formula is executed each time when new call is submitted to the system and this causes that delay appears in setting the connection, even when the traffic is low.

The investigated in the paper approach assumes that for new calls we invoke the AC depending on the current traffic load. The proposed mechanism is based on the following principles (see Fig. 1):

- All requests are accepted when current load (or other specified but representative parameter) is below a given threshold, say thr_1.
- Acceptance of new call demands AC decision when load carried by the link exceeds the threshold thr_1.
- The invocation of AC is again stopped when the load decreases to a given threshold, say thr_2 ($thr_1 \geq thr_2$).

Similar mechanism as above was proposed in [2, 3], for handling TCP flows by implicit AC.

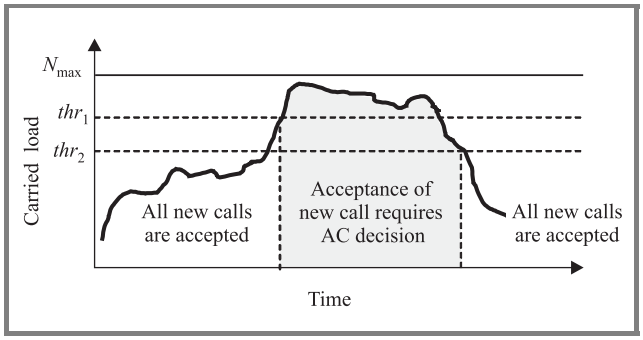


Fig. 1. Areas when the AC function is invoked.

In AC-T strategy, AC function is executed when current load – for simplicity, represented by the number of running connections N – exceeds predefined threshold $trsh_1$. In this case, a new connection is admitted only if:

$$(N+1) \leq N_{\max}, \tag{2}$$

where:

N – number of running connections,

$N_{\max}$ – maximum number of simultaneous connections allowed by the AC.

Assuming that each call requests the effective bandwidth R (homogenous traffic sources case) the $N_{\max}$ is calculated as:

$$N_{\max} = \frac{C}{R}, \tag{3}$$

where:

C – the link capacity.

3. Disadvantages of AC-T

We assume that in proposed approach the AC decision is based on information gathered from measurements. The usual problem of such approach is that accuracy of performing AC strictly depends on credibility of measurement process. In real networks the data for AC function are provided by measurement subsystem after some delay, always greater than zero. Let us name this measurement delay as D. It is likely that in time interval D, there may be a number of connections accepted violating link capacity Eq. (1) or a connection with large bandwidth request may be admitted, consuming or exceeding remaining capacity Eq. (2). So the probability of exceeding link limit $N_{\max}$ is non-zero:

$$\Pr\{N > N_{\max}\} \leq \varepsilon; \quad \varepsilon \geq 0. \tag{4}$$

The objective of AC-T performing is to keep as close to 0 as possible. The best quality is offered to a user if $\varepsilon = 0$. This state may be reached when there is no measurement delay, i.e., $D = 0$. In proposed approaches, where we assume that D is non-zero, the degradation of QoS is expected, as depicted in Fig. 2. The shadowed region corresponds to expected deterioration of running connections quality, e.g., increase of packet losses or packet transfer delay.

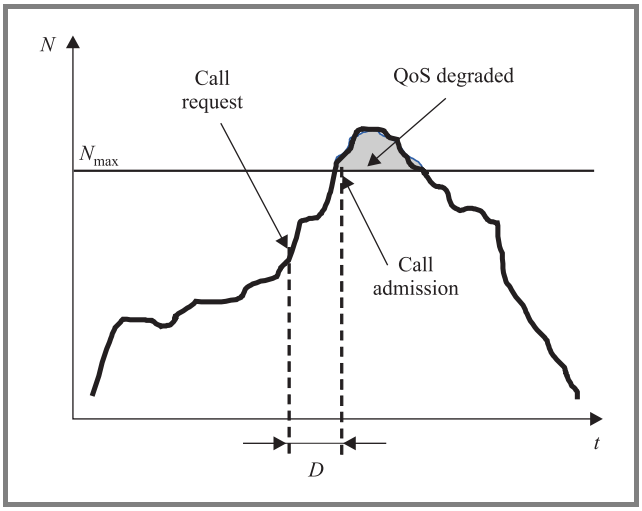


Fig. 2. The impact of measurement delay D on performance of AC function.

4. Simulation model

Let us consider a simple queuing system with single server of capacity C and infinite buffer B. The traffic submitted to this system may come from a number of users each of them requesting specified link bit rates. Assuming that each connection requests bit rate equal to its effective bandwidth R (homogenous case), we get simulation model with $N_{\max}$ available channels each of R bitrate as depicted in Fig. 3, where $N_{\max}$ is calculated, e.g., from Eq. (3).

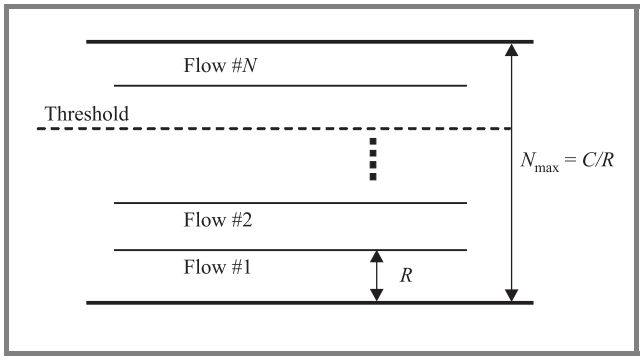


Fig. 3. Simulation model of the link with capacity of $N_{\max}$ connections.

The impact of measurements on AC decision were simulated by introducing the delay parameter D in the simulator. As pointed in the previous section, this parameter reflects the time after which the information about current link load is available for AC function.

5. Effectiveness criteria

Three parameters were taken into account for making an evaluation of the discussed approach for performing AC-T.

Accepted call ratio (*ACR*), defined as a ratio of calls accepted by AC to the total number of submitted calls:

$$ACR = \frac{accepted\_calls}{call\_attempts}. \tag{5}$$

Let us remark that the *ACR* is strictly related to the ratio of successfully completed calls, if we do not consider the number of unfinished calls at the end of the observation time interval.

Now, let us define the QoS experienced by the user as a percentage of connection time when QoS degradation is observed. Mostly, this happens when the number of running connections is greater than $N_{\max}$ and, as a consequence, the offered load is greater than the link capacity. Thus we introduce two next parameters for characterising the QoS degradation level: degradation time ratio and cumulative degradation time ratio.

Degradation time ratio (*DTR*) corresponds to the observation of a single connection duration time t_C and represents the part of t_C when we have more than $N_{\max}$ running connections in the system (jointly with the connection in question). In general, the $N_{\max}$ may be exceeded several times during connection time t_C (Fig. 4).

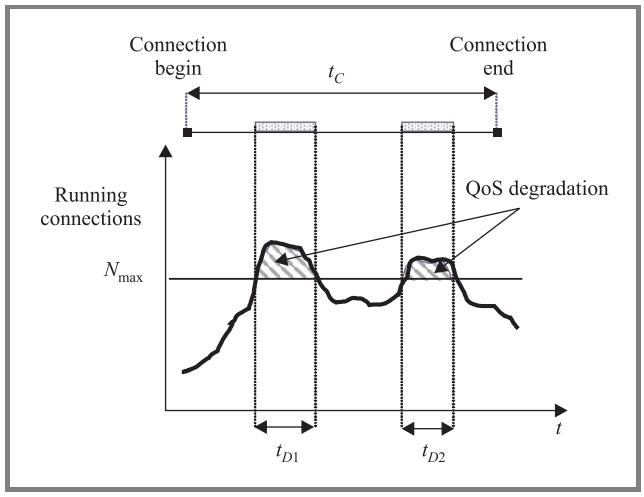


Fig. 4. Time intervals when QoS of the connection is degraded.

Thus, for the jth connection, the *DTR* is calculated as a sum of all time intervals when $N > N_{\max}$, limited for jth connection duration time t_{Cj}:

$$DTR_j = \frac{\sum_k t_{Dk}}{t_{Cj}}; \quad k = 1, 2, \ldots \tag{6}$$

where:

t_{Dk} – kth time interval of jth connection when the number of admitted connections is greater than $N_{\max}$.

Notice that if degradation time is equal or greater than connection duration time then *DTR* becomes 1.

Cumulative degradation time ratio (*CDTR*) is defined as the ratio of time when capacity limit $N_{\max}$ is exceeded to the total time of observation:

$$\frac{t_{N>N_{\max}}}{T}, \tag{7}$$

where:

$t_{N>N_{\max}}$ – time interval when in the system there are more than $N_{\max}$ running connections,

T – time of observation.

The same effectiveness metric as above was proposed in, e.g., [4].

6. Numerical results

In this section we present the exemplary numerical results showing the performance evaluation of the proposed strategy and compare it with the traditional admission control strategy, named AC. The simulation tests were carried

under assumption of Poissonian calls arrival process with rate varied from 50 to 120 calls/s and exponential service time distribution normalised to 1. All calls required the same bandwidth, so the link capacity could be expressed by the maximum number of simultaneously running connections (see Eq. (3)). The capacity allocated for both of strategies was the same. The rest of parameters were set as follows:

– link capacity allocated $N_{\max}$ 100 connections,
– thresholds $trsh_1 = trsh_2 = trsh$,
– measurement delay $D = 0.1$; 1; 5 s.

At first, the comparison of accepted calls ratio versus load was performed. The numerical results of the test are shown in Figs. 5 and 6. The accepted calls ratio differs depending on the chosen strategy. For AC-T the resulting *ACR* factor is approximately the same as we can get from AC. The reason is that AC-T strategy uses accurate information about link load, when the link utilisation exceeds threshold *trsh*, here 0.8 of link capacity.

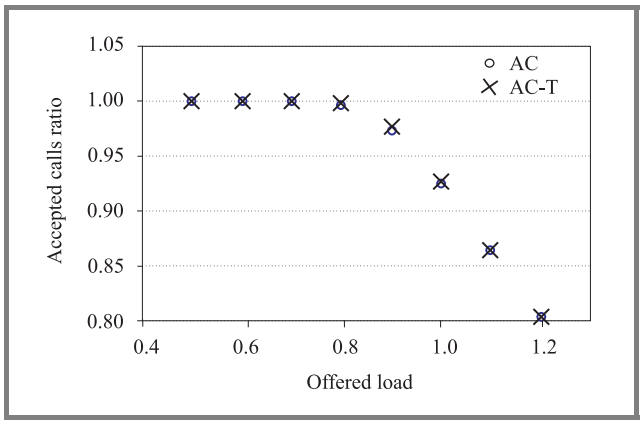


Fig. 5. Accepted calls ratio versus load for $trsh = 0.8$ and $D = 5$ s.

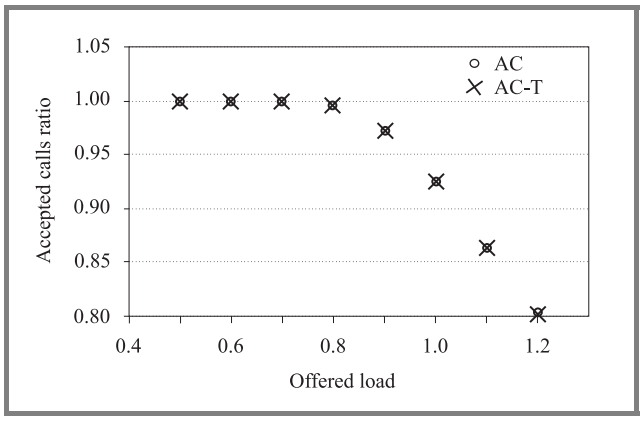


Fig. 6. Accepted calls ratio versus load for $trsh = 0.8$ and $D = 0.1$ s.

Figures 7 and 8 show the impact of threshold value on efficiency of proposed strategies. Notice that for AC-T the *ACR* is close to the received values for accurate AC strategy (see Fig. 7). For more accurate measurements, i.e., smaller *D*, the resulting *ACR* for AC-T is very close to the reference values (see Fig. 8). Comparing Figs. 7 and 8 we can see that there are more admitted connections when *D* increases but with less respect to available link capacity.

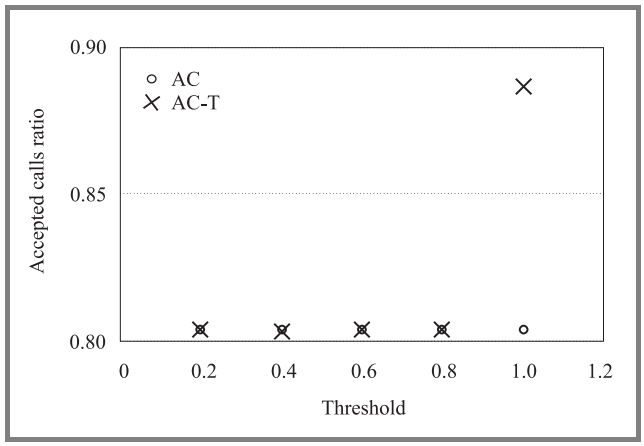


Fig. 7. Accepted calls ratio versus threshold for $\rho = 1.2$ and $D = 5$ s.

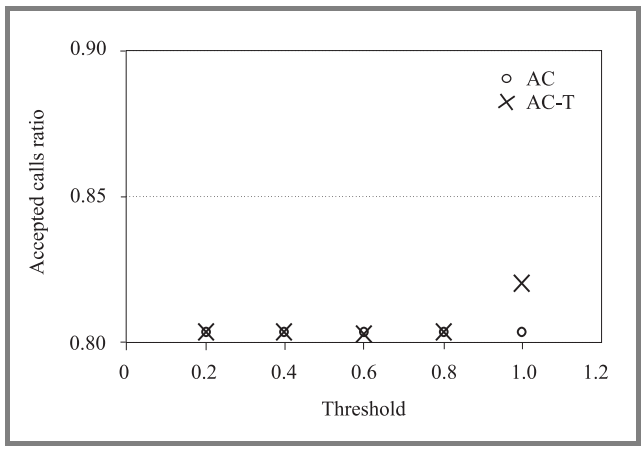


Fig. 8. Accepted calls ratio versus threshold for $\rho = 1.2$ and $D = 0.1$ s.

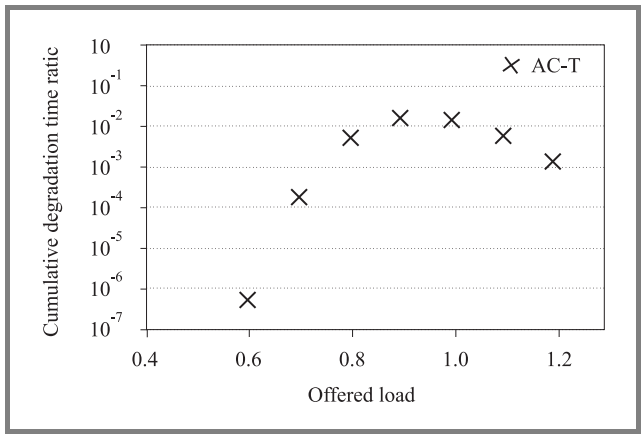


Fig. 9. Cumulative degradation time ratio versus load for $trsh = 0.8$ and $D = 5$ s (note: no events were observed for AC).

Next, the evaluation of *CDTR* was performed. As we could expect, *CDTR* parameter is affected by the accuracy of measurements. The simulation results for inaccurate measurements are depicted in Fig. 9. The resulting *CDTR* is relatively high, but the degradation ratio improves to 10^{-3} when link load exceeds threshold set to $trsh = 0.8$, i.e., the precise AC function is triggered. Note, that degradation concerns all running connections.
In the case of more accurate measurements, for $D = 0.1$ s, we observe negligible degradation level, that is less than 10^{-7} (not shown in the figure).

7. Conclusions

The paper presented and evaluated the approach to facilitate AC function in packet switched networks. It appears that we do not need to invoke the AC each time new call is submitted and we are able to provide QoS guarantees at the assumed level. The effectiveness of proposed AC-T algorithm was evaluated by simulations and compared with hypothetical, accurate AC algorithm, working according to M/M/N/N model.
Anyway, for effective implementation of the strategy AC-T we need to assure that the measurement delay is not too large. The sensitivity of proposed strategies on measurements accuracy can be seen as a trade-off to expected decrease of signaling overhead.
In further work we focus on developing the rule to assign proper threshold values, which seems as a way to manage the measurement issues.

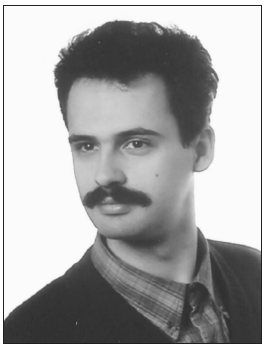
References

[1] S. Blake *et al*., "An architecture for differentiated services", IETF, RFC 2475, Dec. 1998.

[2] S. Ben Fredj, S. Oueslati-Boulahia, and J. W. Roberts, "Measurement-based admission control for elastic traffic", in *Proc. Int. Teletraff. Congr. ITC-17*, Salvador, Brazil, 2001.

[3] N. Benameur, S. Oueslati, J. W. Roberts, "Experimental implementation of implicit admission control", Tech. Rep. 279TD(03)026, COST-279, 2003.

[4] R. Martin, M. Menth, and J. Charzinski, "Comparison of border-to border budget based network admission control and capacity overprovisioning", COST 279, 2003.



Damian Duda was born in Nysa, Poland, in 1973. He received M.Sc. degree in telecommunications from Military University of Technology, Faculty of Electronics, in 1998. He now works as an IT specialist for Military Communication Institute, Zegrze, Poland. His research is focused mainly on modeling, simulation and evaluation of traffic control mechanisms in packet switched QoS networks.
e-mail: d.duda@wil.waw.pl
Military Communication Institute
05-130 Zegrze, Poland

Wojciech Burakowski was born in Poland, in 1951. He received the M.Sc., Ph.D. and D.Sc. degrees in telecommunications from Warsaw University of Technology, in 1975, 1982 and 1992, respectively. He is now a Professor at the Institute of Telecommunications, Warsaw University of Technology. Since 1990, he has been involved in the European projects COST 224, COST 242, COST 257 and 279. His research interests include ATM and IP network design as well as traffic control mechanisms.
e-mail: wb@wil.waw.pl
Military Communication Institute
05-130 Zegrze, Poland

Paper

Performance evaluation of the multiple output queueing switch with different buffer arrangements strategy

Grzegorz Danilewicz, Wojciech Kabaciński, Janusz Kleban, Damian Parniewicz, and Patryk Dąbrowski

Abstract— Performance evaluation of the multiple output queueing (MOQ) switch recently proposed by us is discussed in this paper. In the MOQ switch both the switch fabric and buffers can operate at the same speed as input and output ports. This solution does not need any speedup in the switch fabric as well as any matching algorithms between inputs and outputs. In this paper new performance measures for the proposed MOQ switch are evaluated. The simulation studies have been carried out for switches with different buffer arrangements strategy and of capacity 2×2, 4×4, 8×8, 16×16 and 32×32, and under selected traffic patterns. The simulations results are also compared with OQ switches of the same sizes.

Keywords— packet switching, switching fabric, multiple output queueing.

1. Introduction

The tremendous increase in the speed of data transport on optical fibers has caused a need of deploying next generation network nodes (switches/routers) with high-speed interfaces and large switching capacity. The challenges of building new network nodes include: implementing a large capacity switch fabric providing high-speed interconnection and devising a fast arbitration scheme resolving output contention problems. One of constrains that limits the switching capacity is the speed of memories used for buffering packets to resolve contention resolution in packet switches. Buffers can be placed on inputs, outputs, inputs and outputs, and/or within the switch fabric. Depending on the buffer placement respective switches are called input queued (IQ), output queued (OQ), combined input and output queued (CIOQ) and combined input and crosspoint queued (CICQ) [1].

In [2] we have proposed a new switch architecture which uses multiple output queueing (MOQ). In this architecture buffers are located at output ports and are divided into N separate queues. Each of N queues in one output port stores packets from one input port. We assume, that fixed-length switching technology is used, i.e., variable-length packets are segmented into fixed-length packets, called time slots or cells, at inputs and reassembled at the outputs. We will use terms cell and packet interchangeably further on. In the proposed architecture at most one packet is to be written to the one output queue in one time slot. Therefore, the memory speed is equal to the line speed, but the performance of the switch is very similar to those of OQ switch. The proposed architecture is very promising and looks attractive for constructing high-capacity high-speed packet switches. In this paper new results of simulation studies carried out for the MOQ switch with different buffer arrangements strategy and selected traffic patterns are presented.

The rest of the paper is organized as follows. In Section 2 the general switch architecture proposed in [2] is reminded. In the next section the parameters of simulation researches are explained. Then performance measures for the proposed switch architecture with different buffer arrangements strategy under different traffic patterns are presented and compared with results obtained for OQ. Then we come to conclusions.

2. The switch architecture

The detailed description of MOQ switch architecture is given in [2]. In this switch output queues are located at output side of the switch. To reduce the memory speed an output buffer at each output port is divided into N separate queues. Each queue stores packets directed to the output only form one input. In this way this architecture is similar to the virtual output queueing (VOQ) switch, but multiple buffers are located at output ports not at input ports. The general architecture of the switch is shown in Fig. 1. The switch consists of N input ports, N output ports and the switch fabric. Input and output ports can be implemented on separated ingress and egress cards or they may be placed on one line card. At the output port the buffer memory is divided into N separate queues. Each queue stores packets directed from one input port. The output queue denoted by $OQ_{j,i}$ at the output port j stores packets directed to this output port from input i. At one time slot each input port can send at most one packet and each output port can receive up to N packets, each from different input ports. Therefore, these packets can be simultaneously written to N queues.

The main advantage of these architecture is that it can operate at the same speed as input and output ports, and the lack of arbitration logic, which decides which packets from inputs will be transferred through the switch fabric to output ports (this arbitration mechanism is needed

in VOQ switches). However, since we have N queues in each output port, it is necessary to use an output arbiter, which chooses a packet to be sent to the output line. We propose to use round-robin scheme, which is widely used because of its fairness. The switch fabric in the proposed switch should have a capacity of $N \times N^2$ and should be nonblocking at the packet level.

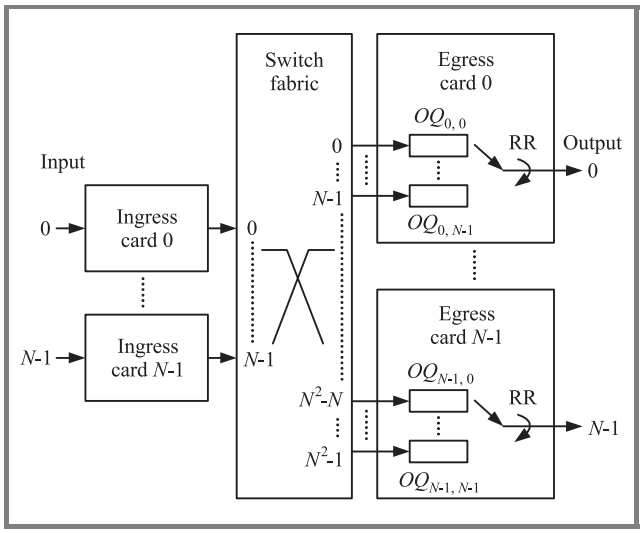


Fig. 1. The switch architecture with multiple output queueing.

In [2] we have shown that the MOQ switch has the hardware and wiring complexity similar to the VOQ switch, but for uniform traffic its performance is very similar to the OQ switch. In this paper more research results are given for uniformly and non-uniformly distributed traffics.

3. Buffer arrangements

Buffers in output ports are arranged into N separate queues. When N packets from N input ports are directed to one output port in the same time slot, each packet is written to the different queues. Therefore, the memory speed is the same as the line speed. Buffers may be arranged as separate queues with independent write pointers or as a memory bank with one pointer which points the same memory cells in each queue. The implementation of this two buffer arrangements strategies in hardware can influence the costs of MOQ switch.

Packets from N queues in each output port are read out using the round-robin algorithm. When independent write pointers are used, the round-robin pointer, denoted by RR (Fig. 1), is moved to the queue next to those read out in the previous time slot. When packets are written to the same position of the buffers (one write pointer is used), the operation of RR is modified in such a way, that when all packets from the same position are already read out, the RR is set back to 0. The operation of these two arrangements will be described by means of the following example.

In the first case the separate pointer is assign to each queue. This pointer, denoted by $MP_{j,i}$, points the end of queue $OQ_{j,i}$, where the next incoming packet to output j from input i will be written to. The example for output x is shown in Fig. 2. It is assumed that all queues are empty at the beginning of the first time slot. Pointers are shown by arrows which show the state of the pointers at the end of respective time slots. In the first time slot two packets (numbered 1 and 2) from inputs 0 and 1 arrive to the considered output x. The round-robin pointer (RR) is set to 0 (the head of line blocking (HOL) packet from $OQ_{x,0}$ has the highest priority). Since buffer $OQ_{x,0}$ is empty, the packet from input 0 is immediately directed to the output, the RR pointer is set to 1, and packet 2 is stored in $OQ_{x,1}$. The state of RR at the end of the time slot is shown in Fig. 2. The pointer of $OQ_{x,1}$ is moved to the next memory cell. In the next time slot packets from inputs 0, 1 and 3 arrive (numbered 3, 4, and 5, respectively). They are stored in respective queues, while packet 2 from $OQ_{x,1}$ is sent out. During the third time slot packets 6, 7 and 8 arrive from inputs 1, 2, and 3, respectively. Since RR is now set to 2 and buffer $OQ_{x,2}$ is empty, packet 7 is sent directly to the output, while packets 6 and 8 are stored in $OQ_{x,1}$ and $OQ_{x,3}$. In the next time slot packet 5 will be sent out from $OQ_{x,3}$. The sequence of packets from the same input port is preserve.

In the second case there is one pointer for all queues. This pointer, denoted by MP_j, points to the memory cells in all queues of output j, where the next incoming packets will be written to. The example is shown in Fig. 3. In the first time slot two packets (numbered 1 and 2) from inputs 0 and 1 arrive to the considered output x. The round-robin pointer (RR) is set to 0 (the HOL packet from $OQ_{x,0}$ has the highest priority). Since buffer $OQ_{x,0}$ is empty, the packet from input 0 is immediately directed to the output, packet 2 is stored in $OQ_{x,1}$, the MP_x is moved to the next memory cells in all queues (shown by arrows in Fig. 3), and the RR pointer is set to 1 (here also the state of RR is shown et the end of the time slot). In the next time slot packets from inputs 0, 1 and 3 arrive (numbered 3, 4, and 5, respectively). They are stored in the second memory cell of respective queues, while packet 2 from $OQ_{x,1}$ is sent out. After this packet is read out, there is no any packet in the first memory cell in all queues. Therefore, the next cells in the queues are moved to the HOL position, and the RR is set to 0. During the third time slot packets 6, 7 and 8 arrive from inputs 1, 2, and 3, respectively. Since RR is now set to 0, packet 3 from $OQ_{x,0}$ is sent to the output, while new packets are written to the buffer. In the next three time slots packets 4, 5, and 6 will be sent out from $OQ_{x,1}$, $OQ_{x,3}$, and $OQ_{x,1}$, respectively.

In this second approach all packets which arrive to the given output are written in the same position of each buffer. So we can use only such positions where all memory cells are empty. When in the given time slot less than N packets arrive to the output, some memory cells will be empty and they could not be used to store packets until all packet in the same position of all buffers are read out. There-

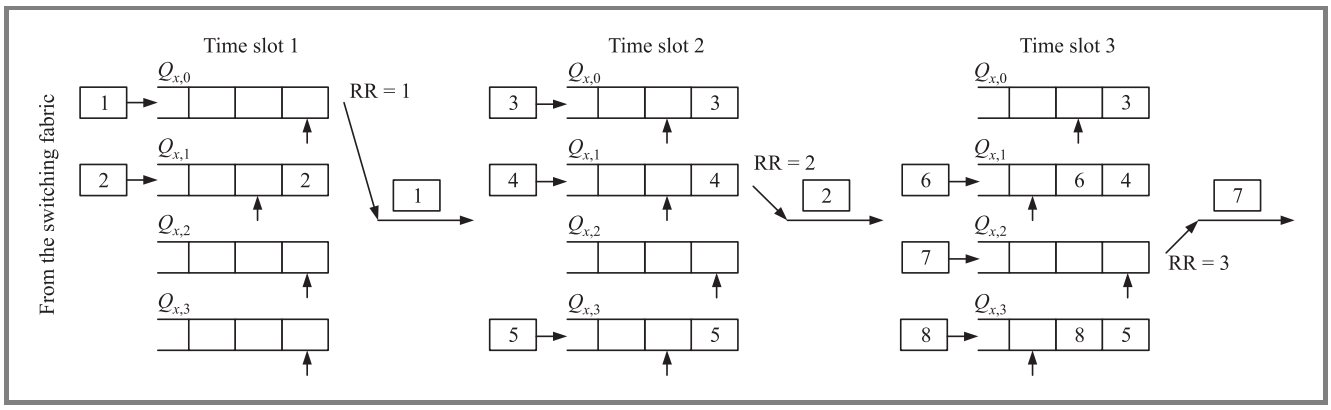


Fig. 2. The example of buffer operation with separate pointers.

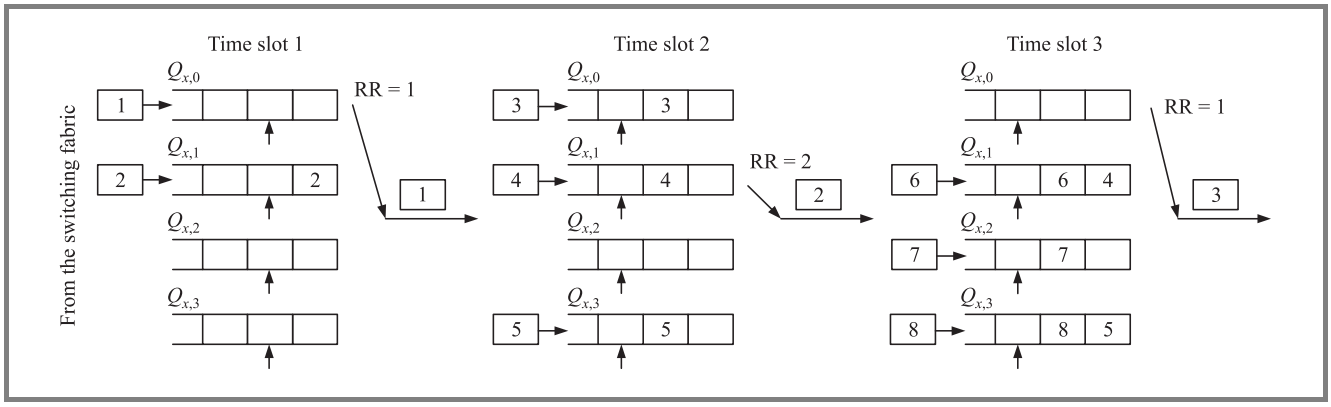


Fig. 3. The example of buffer operation with one pointer.

fore, the memory is not used as efficiently as in the first approach.

4. Simulation experiments

4.1. Packet arrival models

The Bernoulli arrival model is considered in the paper. In this arrival model cells arrive at each input in a slot-by-slot manner. Under Bernoulli arrival process, the probability that there is a cell arriving in each time slot is identical and is independent of any other slot. The probability that cell may arrive in a time slot is denoted by p and is referred to as the load of the input [3]. This kind of traffic defines a memoryless random arrival pattern.

4.2. Traffic distribution models

We consider several traffic distribution models which determines the probability that a cell which arrive in an input will be directed to the certain output. The considered traffic models are:

Uniformly distributed traffic – this type of traffic is the most commonly used traffic profile test in the literature [4–6]. In a uniformly distributed traffic probability p_{ij} that packet from input i will be directed to output j is uniformly distributed through all outputs, i.e.,

$$p_{ij} = p/N \qquad \forall i, j. \tag{1}$$

Non-uniformly distributed traffic – in this traffic model some outputs have a higher probability of being selected, and respective probability p_{ij} was calculated according to the following equation [1]:

$$p_{ij} = \begin{cases} \dfrac{p}{2} & \text{for} \quad i = j \\ \dfrac{p}{2(N-1)} & \text{for} \quad i \neq j \end{cases}. \tag{2}$$

Diagonally distributed traffic – in this model the traffic is concentrated in two diagonals of the traffic matrix, and the probability that a packet will be directed to any of the two outputs is equal to $p/2$ [4–7]. This loading is skewed in the sense that input i has packets only for outputs i and $|i+1|$, where $|k|$ denotes the modulo N operation ($|k| = k \bmod N$).

Log-diagonally distributed traffic – for a log-diagonally distributed traffic, the traffic matrix is defined by equation [4, 6]:

$$p_{ij} = 2p_{i|j+1|} \tag{3}$$

and

$$\sum_i p_{ij} = p. \tag{4}$$

For example, the load distribution at input 1 across outputs is

$$p_{1j} = 2^{N-j} p/(2^N - 1). \quad (5)$$

Lin-diagonally distributed traffic – this traffic is a further modification of diagonally distributed traffic. Lin-diagonally distributed traffic can be defined as

$$\overline{p}_d = p\,(N-d)\big/(N(N+1)/2) \quad (6)$$

with $d = 0,...,N-1$, then $p_{ij} = \overline{p}_d$ if $j = |i+d|_N$. This traffic model is an intermediate case between the uniformly and log-diagonally distributed traffics in which the load decreases linearly from one diagonal to the other [8].

5. Performance evaluation

In this section performance evaluation of the MOQ switch will be presented and compared with OQ switches. Simulation results indicate that mean time delay (MTD in time slots) and cell lose probability for MOQ switches with different buffer arrangements strategy are very similar and differences are unnoticeable. Thus only results for separate pointers are presented. The results have been obtained for switches of different sizes: 2×2, 4×4, 8×8, 16×16, and 32×32. The results for switches of different sizes are very similar in shapes. The adopted buffer size assures — for each value of traffic load — stable values of MTD (the application of larger buffers do not lead to increase this waiting time). For OQ switch we have assumed that the buffer size is infinity. Results for OQ will be presented only for Bernoulli arrivals and uniformly distributed traffic and will be obtained from calculations based on formulas given in [3].

In Fig. 4 different traffic distribution models are compared in 16×16 MOQ switch, where buffer size is equal to 16. The highest MTD in MOQ switch is observed for uniformly distributed traffic. This kind of traffic gives similar MTD values for MOQ and OQ switches but MOQ is slightly better.

The MTD in MOQ switches of different sizes versus p for uniform traffic and $L = 16$ is plotted in Fig. 5. It can be seen that when the switch size is growing the MTD also grows but very slowly. This delay is almost the same as the theoretical MTD calculated for OQ switches of the same capacities.

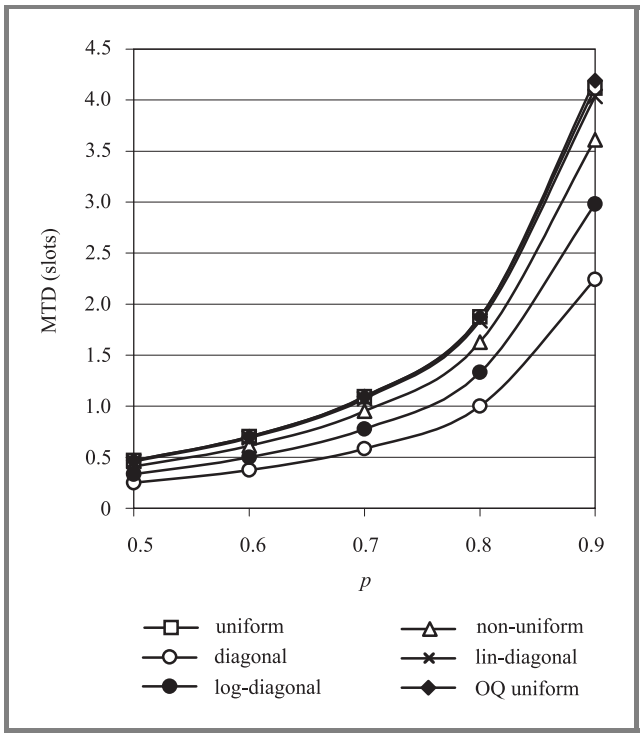


Fig. 4. The MTD for Bernoulli arrivals with different distributed traffic in a 16×16 switch with MOQ ($L = 16$) and OQ ($L =$ infinity).

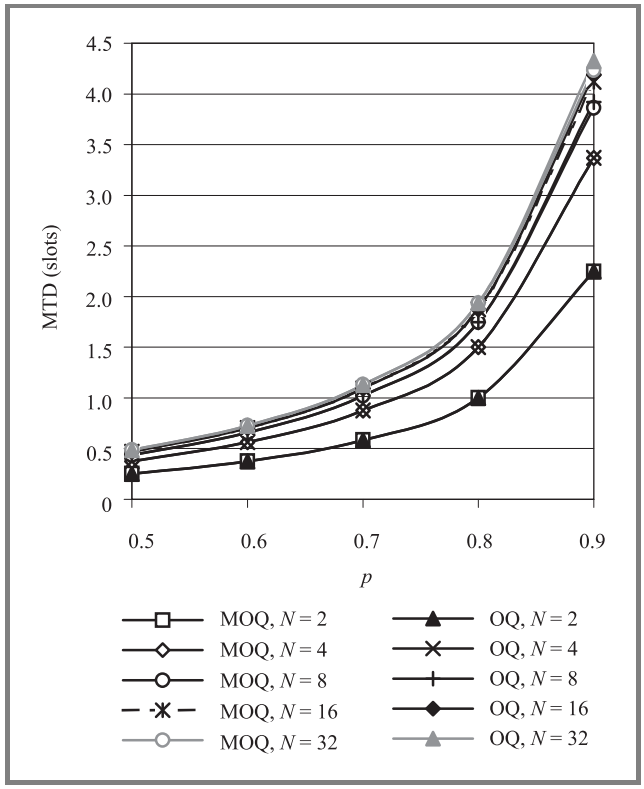


Fig. 5. The MTD for Bernoulli arrivals with uniformly distributed traffic and for different capacities of MOQ ($L = 16$) and OQ switches ($L =$ infinity).

Another important performance measure for packet switches is the cell loss probability (CLP). Figure 6 compares CLP obtained for MOQ switch with the results calculated for the OQ switch. CLP for OQ switch is calculated from formula CLP $= 1-(\rho_0/p)$, where p is the offered load. Proof of this formula can be found in [3]. It is intuitively clear, that the proposed switch architecture requires greater total number of memory cells (N buffers for each output port) in order to keep the same value of CLP parameter as in the case of switches with single output queue for

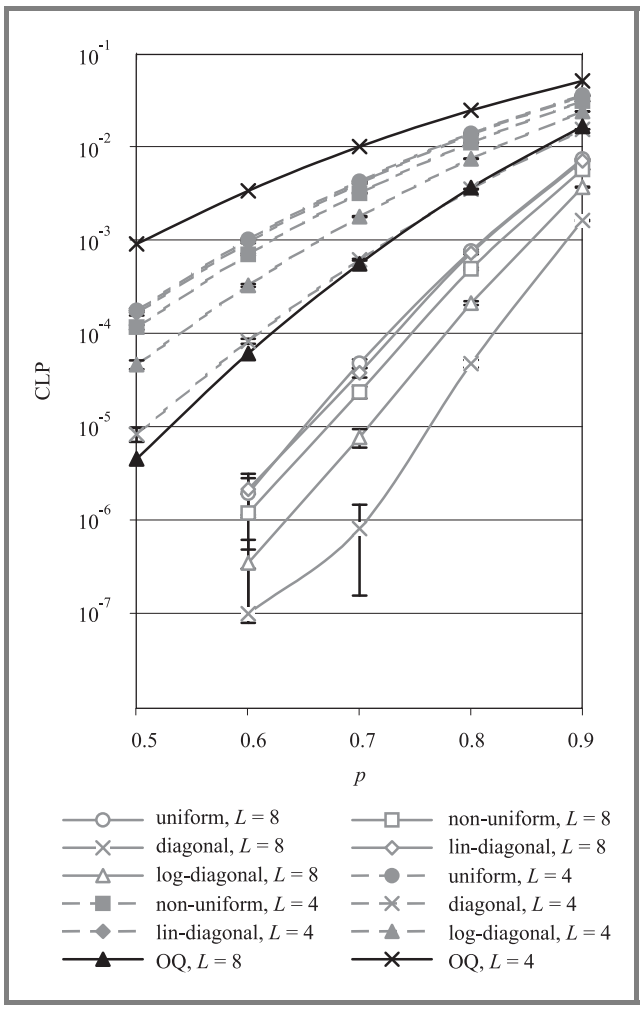


Fig. 6. The CLP for Bernoulli arrivals with different distributed traffic and for different lengths of buffers in MOQ and OQ 16×16 switches.

each output port. From Fig. 6 it can be seen that when the length of each buffer is equal to the same value (it means that in OQ we use L memory cells for one output while in MOQ we use $N \times L$ memory cells for one output) then CLP in MOQ switch is about one order of magnitude better than CLP in OQ switch. CLP for long buffers ($L = 16$) is practically unnoticeably in our simulations.

6. Conclusions

We have presented the packet switch architecture which uses multiple output queueing and its performance under different traffic patterns. Our studies lead us to conclusion that buffer arrangements strategy is only important for the practical switch implementation not for performance of the switch fabric. The hardware complexity of MOQ architecture is very similar to VOQ switch but its performance is very comparable to OQ switch. The MTD is the same for both MOQ and OQ architectures for uniformly distributed traffic. The CLP is better for MOQ than for OQ, however, N times more memory cells are used in the MOQ switch architecture. The MOQ architecture is also very promising since it can naturally support multicast traffic. It should be also noted, that the MOQ switch can be also modified to support different traffic priorities. Each of N output buffers of each output port can be further divided to support different packet priorities. Evaluation of such switch architecture is also the subject of further studies.

References

[1] K. Yoshigoe and K. J. Christensen, "An evolution to crossbar switches with virtual ouptut queueing and buffered cross points", *IEEE Network*, vol. 17, no. 5, pp. 48–56, 2003.

[2] G. Danilewicz, M. Głąbowski, W. Kabaciński, and J. Kleban, "Packet switch architecture with multiple output queueing", in *6th NATO Reg. Conf. Milit. Commun. Inform. Syst. RCMCIS 2004*, Zegrze, Poland, 2004.

[3] H. J. Chao, C. H. Lam, and E. Oki, *Broadband Packet Switching Technologies: A Practical Guide to ATM Switches in IP Routers*. New York: Wiley, 2001.

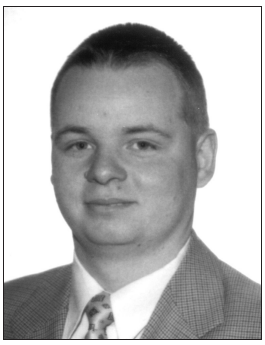
[4] P. Giaccone, D. Shah, and S. Prabhakar, "An implementable parallel scheduler for input-queued switches", *IEEE Micro*, vol. 22, no. 1, pp. 19–25, 2002.

[5] D. Shah, P. Giacconeand, and B. Prabhakar, "Efficent randomized algorithms for input-queued switch scheduling", *Proc. Hot-Intercon. IX*, vol. 22, no. 1, pp. 10–18, 2002.

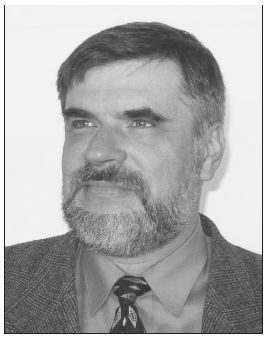
[6] P. Giaccone, B. Prabhakar, and D. Shah, "Randomized scheduling algorithms for high-aggregate bandwidth switches", *IEEE J. Select. Areas Commun.*, vol. 21, no. 4, pp. 546–559, 2003.

[7] Y. Jiang and M. Hamdi, "A fully desynchronized round-robin matching scheduler for a VOQ packet switch architecture", in *IEEE HPSR'01*, Dallas, USA, 2001, pp. 407–411.

[8] A. Bianco, P. Giaccone, E. Leonardi, and F. Neri, "A framework for differential frame-based matching algorithms in input-queued switches", in *IEEE INFOCOM'04*, Hong Kong, 2004.



Grzegorz Danilewicz was born in Poznań, Poland, in 1968. He received the M.Sc. and Ph.D. degrees in telecommunications from the Poznań University of Technology (PUT), Poland, in 1993 and 2001, respectively. Since 1993 he has been working in the Institute of Electronics, Poznań University of Technology, where he currently is an Assistant Professor. His scientific interests cover photonic broadband switching systems with special regard to the realization of multicast connections in such systems. He is a member of the IEEE Communication Society. He has published one book and 35 papers.
e-mail: G.Danilewicz@et.put.poznan.pl
Institute of Electronics and Telecommunications
Poznań University of Technology
Piotrowo st 3A
60-965 Poznań, Poland



Wojciech Kabaciński received the M.Sc., Ph.D., and D.Sc. degrees in telecommunications from the Poznań University of Technology (PUT), Poland, in 1983, 1988 and 1999, respectively. Since 1983 he has been working in the Institute of Electronics and Telecommunications, Poznań University of Technology, where he currently is an Associate Professor. His scientific interests cover broadband switching networks and photonic switching. He has published three books, over 100 papers and has 10 patents. Professor Kabaciński is a member of the IEEE Communication Society and the Association of Polish Electrical Engineers.

e-mail: Wojciech.Kabacinski@et.put.poznan.pl
Institute of Electronics and Telecommunications
Poznań University of Technology
Piotrowo st 3A
60-965 Poznań, Poland

Janusz Kleban was born in Pobiedziska, Poland. He received the M.Sc. and Ph.D. degrees in telecommunications from the Poznań University of Technology (PUT) in 1982 and 1990, respectively. From August 1982 to November 1983 he was with Computer Centre for Building Industry in Poznań, where he worked on data transmission systems. He has been with Institute of Electronics and Telecommunications at PUT, where he currently is an Assistant Professor, since December 1983. He is involved in research and teaching in the areas of computer networks, switching networks, broadband networks and various aspects of networking. He is author and co-author of many publications and unpublished reports.

e-mail: jkleban@et.put.poznan.pl
Institute of Electronics and Telecommunications
Poznań University of Technology
Piotrowo st 3A
60-965 Poznań, Poland

Regular paper

The signal to noise ratio in the differential cryptanalysis of 9 rounds of data encryption standard

Michał Misztal

Abstract— There is presented the differential cryptanalysis method of attack on data encryption standard (DES) reduced to 9 rounds. More precise estimation than that of Biham and Shamir of the signal to noise (S/N) ratio is obtained. Also, method of using the ratio in calculation of required number of plaintexts is suggested. There are given results (time of performance) and implementation's issues of practical realisation of this attack.

Keywords— block cipher DES, differential cryptanalysis, substitution boxes, S/N ratio.

1. Introduction

Differential cryptanalysis and its modifications (like for instance impossible differentials) are the most powerful method of attacks on the popular symmetric cryptosystems – block ciphers. The idea of differential cryptanalysis was introduced in 1991 and applied to the former data encryption standard (DES). At present every newly designed block cipher must be at once evaluate due to resistance to the differential attack. Hence differential attacks on contemporary block ciphers are possible only in theory or for small number of rounds [1]. However differential cryptanalysis could be still improved by applying in practice to some well-known ciphers like mentioned DES.

This article is the continuation of paper [2], in which practical attack with differential cryptanalysis on DES cipher reduced to 8 rounds was performed. In the introduction of that paper it was stated that attack on more than 6 or 8 rounds of DES requires too much amount of data (encryption of too many plaintexts) to be preformed in practice. Due to increasing of computational power of computers (processor speed, capacity of operational and disc memory) attacks which were considered as only theoretical become now possible in practice. At the beginning of 90's, when idea of differential cryptanalysis was born, its inventors did not have possibility to verify their theories in practice. Simple attacks on 3 or 4 rounds of DES were possible but for more rounds only theoretical estimation of required amount of data and complexity time was done. Practical attack on 8 rounds of DES [2] showed that many of these theoretical estimations differ from reality (for example 25 000 pairs is far to small to succeed). At presence thanks to available processors and especially to capacity of memory (cf. Section 5) we can perform attack even on 9 rounds. Thanks to that we can do some experiments and obtain practical, precise results and than compare them to theory. It is the main aim of this paper.

We start from the brief recollection of idea of differential cryptanalysis and the scheme of DES cipher. More details can be found in given references [2–4]. In Section 4 we show how to calculate the S/N ratio of counting scheme of attack on 9 round of DES in more precise way than up to now. We also suggest a method of using the ratio in calculation of required number of plaintexts. We theoretically calculate the efficacy of filtration of wrong pairs and compare it to practice. In Section 5 the implementation's issues of practical realisation of this attack are given. At the end we present results (running time and efficacy) of performed cryptanalysis.

2. The DES algorithm

The data encryption standard algorithm was the encryption standard since year 1977. In year 2001 it was replaced by chosen in contest block cipher Rijndael, which became advanced encryption standard (AES).

The DES algorithm is a block cipher, which in standard version encrypts 64-bit block of plaintext to 64-bit block of ciphertext with 64-bit key. Due to standard actual length of key is only 56-bits and 8 bits are extra bits used only for parity check. Algorithm consists of 16 rounds and is based on structure called Feistel's network. In every round left half of block is xored with result of f function applied to right half of block. Then in every round but last both halves are swapped. Hence f function is main element of every round. It transforms half of encrypted block (32-bits) with 48-bit subkey of round. Every subkey is obtained from main key by algorithm called key schedule.

The f function uses two permutations E and P and also 8 nonlinear mappings called substitution tables or substitution boxes (briefly s-boxes). Extending permutation E transforms 32-bit block to 48-bit block. Permutation P transforms 32-bit block to 32-bit block. In every s-box 6-bit input is transformed to 4-bit output. General scheme of algorithm and scheme of f function are presented in Figs. 1 and 2. Key schedule, permutations E and P and all eight s-boxes can be found in given references.

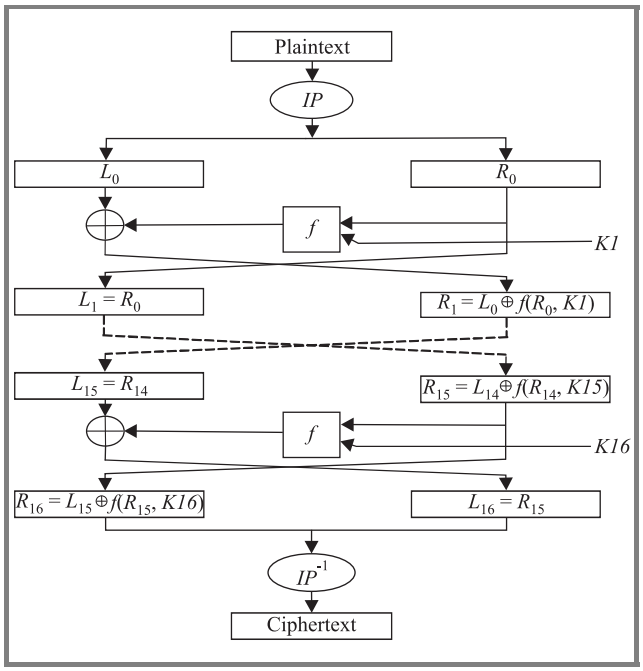


Fig. 1. Scheme of DES algorithm.

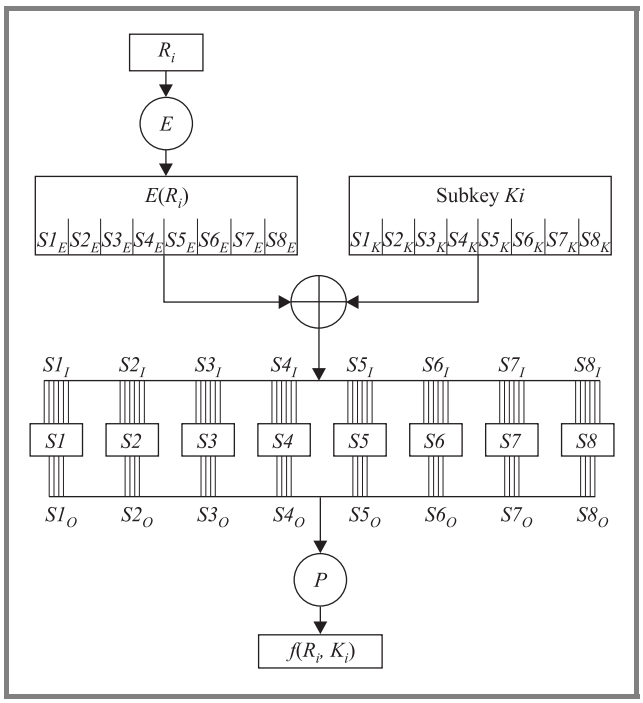


Fig. 2. Scheme of f function.

The DES algorithm reduced to n rounds is an algorithm in which two changes were made:

- number of rounds was reduced from 16 to n, but in the nth (last) round there is no swapping of halves, like after 16th round in standard version;
- permutations IP and IP^{-1} were removed due to their insignificancy to cryptanalysis.

3. Differential cryptanalysis

The differential cryptanalysis was introduced in year 1991 by Biham and Shamir [3] as modern method of cryptanalysis of DES. At least in theory this method is better than exhaustive search, it means testing all possible 2^{56} keys. It is based on dependency between pairs of plaintexts with certain difference (in term of XOR) and differences of their ciphertexts. From above the name "differential" was derived. It is a chosen plaintext (CPA) type of cryptanalysis.

Basic idea of differential cryptanalysis is observation of behaviour of pair of blocks with certain difference transformed through rounds of cipher rather than single block. For linear mappings like permutations, XOR operations the difference of pair of blocks behaves in deterministic way, like single block. The most important thing is that XOR with unknown key does not change this difference. If two arbitrary blocks X and X^* with known difference $X' = X \oplus X^*$ are XORed with unknown key K then the new values of this blocks $X \oplus K$ and $X^* \oplus K$ are unknown but their difference X' remains the same. It happens because of property of XOR operation in which $K \oplus K = 0$. The only problem during analysis of propagation of differences is the application of nonlinear mappings, i.e., s-boxes. For s-boxes we can find input and output difference, which occurs more frequently than others, i.e., with higher probability. However it makes that differential cryptanalysis is only probabilistic method. Its results depend on certain value of key, chosen plaintext and it requires sufficiently many tries to find correct key. To find the best input – output difference propagations of s-boxes the so-called XOR profiles are constructed. The XOR profile of s-box is a table, which shows how many certain input difference goes to certain output difference. XOR profiles are discussed also in the next section.

If we know or we can predict behaviour of differences in individual operations and rounds, then we can find input difference, i.e., difference of two plaintexts which after first round goes to certain difference with some probability, which subsequently after second round goes to another difference with some probability and so on. This sequence of successive differences between successive rounds from plaintext difference to ciphertext difference is called differential characteristic. Every characteristic has its probability, which is calculated as product of probabilities of difference propagations for all rounds. Main problem in differential cryptanalysis is to find "good" differential characteristic, which means characteristic with high probability.

Only differential characteristic with sufficiently high probability makes possible to perform a differential attack. If we have such characteristic we choose pairs of plaintexts with difference given by this characteristic and we obtain their ciphertexts. Then we try to discard pairs, which do not follow our characteristic. This process we call filtration. We know only plaintext and ciphertext difference and we do not

know difference between individual rounds, because cipher is a black box for us. Hence we cannot be sure which pairs are good, i.e., follow our characteristic and which pairs are wrong and only look like they follow this characteristic. In the filtration process we can analyse ciphertext difference and discard pairs, which are wrong for sure. All good pairs will survive this filtration but some number of wrong pairs will survive as well.

After filtration for every non-discarded pair we may find possible subkeys for example last or first round or parts of these subkeys (for example 30 out of 48 bits) in procedure called key recovery. Detailed scheme of this procedure can be found in [3] or [2]. Every pair suggests several subkeys. Good pair suggests exactly one good subkey and few wrong subkeys. Wrong pair suggests only wrong subkeys. Hence we have to count for every non-discarded pair how many times every subkey occurred. For sufficient number of analysed pairs the correct subkey should be the most frequently appeared subkey. The indicator of how many times the correct subkey is more frequent than other subkeys is signal to noise (S/N) ratio. Precise calculation of this parameter can inform us about chances of success of certain attack. Due to this parameter we can also determine the number of pairs required to assure success of the attack. In the next section we discuss how to calculate the S/N ratio in attack on 9 rounds DES and how to determine the required number of pairs from the S/N ratio. Given method is general and could be used in other attacks and for other ciphers.

4. The signal to noise ratio in attack on 9 rounds of DES

4.1. Differential characteristic and its probability

To attack algorithm DES reduced to 9 rounds we use the following 6-round differential characteristic – see Fig. 3.

The characteristic is taken from [3] and it is the best differential characteristic of DES found up to now. Its probability is a product of probabilities of 6 successive rounds and it is equal to:

$$p = \frac{(12\cdot 14\cdot 16)\cdot 1\cdot (10\cdot 16)\cdot 1\cdot (10\cdot 16)\cdot 1}{64^3\cdot 4\cdot 64^2\cdot 64^2\cdot 4} = \frac{2^{17}\cdot 525}{2^{46}}$$

$$= \frac{525}{2^{29}} \approx 9.7788870334625244140625\cdot 10^{-7}$$

$$\approx \frac{1}{1000000}\,.$$

With this characteristic we can attack 9 rounds of data encryption standard by adding three more rounds (so-called 3R attack). Due to the characteristic for good pairs we have:

$$R'_6 = 40\ 5C\ 00\ 00_x\,,$$

hence for five s-boxes: $S2$, $S5$, $S6$, $S7$, $S8$:

$$S'_{E7} = S'_{I7} = 0 \quad \text{and} \quad S'_{O7} = 0,$$

where:

S'_{E7} means 6-bit difference after permutation E in 7th round,

S'_{I7} means 6-bit difference before s-boxes layer in 7th round,

S'_{O7} means 4-bit difference after s-boxes layer in 7th round.

Due to scheme of DES for these s-boxes we have the following relation:

$$f(R_8, K9)' = C'_L \oplus R'_5 \oplus f(R_6, K7)\,.$$

The characteristic allows us to obtain for these 5 s-boxes input and output in 9th round required for key recovery procedure. By applying the characteristic and the key recovery procedure for one round (9th in that case) we obtain 5 (s-boxes) · 6 bits = 30 bits of subkey $K9$, and 30 bits of main key as well. Remaining 26 bits of main key can be found by exhaustive search.

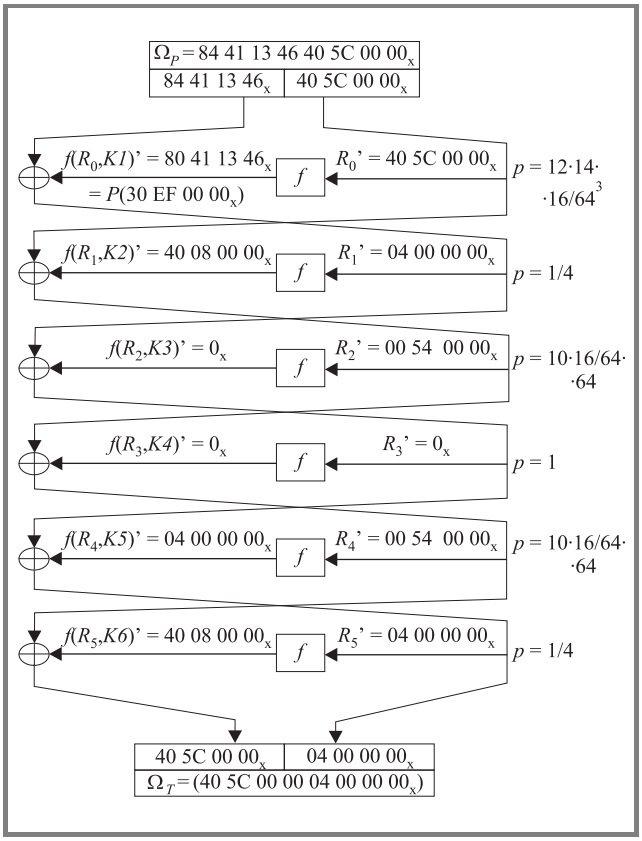


Fig. 3. Six-round characteristic.

Due to the probability of the characteristic only some fraction of pairs will follow it. We know only the difference of

plaintexts and corresponding difference of ciphertexts after 9 rounds so we cannot confirm whether it is a good or bad pair. But we can do some analysis for s-boxes $S2$, $S5$, $S6$, $S7$ and $S8$. If for any s-box obtained input difference does not go to obtained output difference, or equivalently set of suggested subkeys is empty, than we know for sure that analysed pair does not follow the characteristic and should be discarded. Some number of wrong pairs cannot be revealed and discarded in that way. Unfiltered pairs will suggest only wrong subkeys and they will provide only disinformation noise. Also good pairs will suggest a few wrong subkeys and exactly one correct subkey. To distinguish the correct subkey from the noise we count occurrences of all suggested subkeys. The correct subkey should occurs more times than others wrong subkeys. The question is how many pairs we need to analyse to distinguish the correct subkey from the noise. To determine the number of required pairs the S/N ratio is introduced. The parameter estimates the ratio of the number of good pairs equals the number of occurrences of the correct subkey (signal) to the number of occurrence of all subkeys (noise).

4.2. The signal to noise ratio parameter

The S/N ratio of counting scheme is defined as ratio of the number of good pairs and average number of counts of wrong subkeys in counting scheme. In other words the parameter shows how many more times will the correct subkey occur than any other subkey. The formula for the S/N ratio is given below:

$$S/N = \frac{mp}{m\alpha\dfrac{\beta}{2^k}} = \frac{2^k p}{\alpha\beta},$$

where:

p – the probability of the characteristic,

k – the number of bits of counted subkeys,

α – the average number of subkeys suggested by one analysed pair,

β – the ratio of analysed pairs to all pairs, an efficiency of filtration,

m – the number of decrypted pairs.

From the formula it is easy to see that:

- the S/N ratio is independent of the number of pairs used in the attack,
- different schemes of counting based on the same characteristic but counting different number of bits of subkey have different value of the S/N ratio.

The number of good pairs required to find the correct subkey is a function of the S/N ratio parameter. For one s-box of DES we assume $k = 6$, $\alpha = 4$ (average number of 6-bits subkeys suggested by one pair), $\beta = 0.8$ (average percentage of possible difference transitions in s-box).

With these values we can calculate the S/N ratio for the attack on 9 rounds in the following way. We use key recovery procedure for 5 s-boxes in the last round simultaneously; hence we assume following values:

- $k = 5 \cdot 6 = 30$ bits,
- $\alpha\beta = 4 \cdot 4 \cdot 4 \cdot 4 \cdot 4 = 4^5 = 2^{10} = 1024$, the average number of 30-bit subkeys suggested by one analysed pair; for every s-box we have in average four 6-bit subkeys, hence to obtain 30-bit subkey for five s-boxes we have to determine all combinations of these 6-bit subkeys.

The parameters α and β could be determined separately as we will do later in the precise estimation of the S/N ratio, and now we only determine their product like above. It is easier now and as we will show it is also precise.
We know the probability of the characteristic and above values of parameters so we can calculate the S/N ratio now:

$$S/N = \frac{2^{30} \cdot \dfrac{1}{1000000}}{4^5} = \frac{2^{20}}{1000000}$$

$$= \frac{1048576}{1000000} = 1.048576 \approx 1\,.$$

It was assumed in [3], that if the S/N is between 1 and 2, then about 20–40 good pairs are sufficient. If the S/N is high then only a few occurrences of right pairs are needed to uniquely identify a right value of the subkey bits. If the S/N is small the number of required pairs is big and when S/N is less than 1 we never find the correct subkey. In that case the correct subkey occurs more rare than other subkeys in average. Hence the maximum value in counter is not the value of the correct subkey even for huge number of pairs. In that case attack would be impossible.
In our attack the S/N is small but higher than 1, what makes the attack possible at least in theory. We have to determine the number of pairs needed to success of the attack. According to [3] 30 good pairs are sufficient. Good pair appears statistically one time per every 1 000 000 generated pairs, hence about 30 million pairs will be needed to perform the attack and to uniquely identify a right value of 30-bit subkey. Experiments have shown (cf. Section 6) that above number of pairs is too small in general. Sometimes that number is sufficient but it happens too rarely. We would like to determine the number of needed pairs, which is sufficient to uniquely identify a right subkey in all cases. 30 million, it is too small for sure. How many pairs we need and how to determine this number in the clear and faultless way? We will try to answer the question and we will start from precise calculation of the S/N ratio.

4.3. Precise calculation of the signal to noise ratio

To determine the number of pairs needed to success of the attack first we have to calculate the S/N ratio in the most

precise way. If this parameter is much higher than 1 a few (3–4) pairs would be sufficient like in the attack on 4 or 6 rounds DES. But in the case of attack on 9 rounds DES the ratio is only slightly higher than 1, so we have to calculate it very precise. We will of course use the formula given previously:

$$S/N = \frac{2^k p}{\alpha\beta}.$$

Value of k remains unchanged: $k = 30$ but the probability p is now exact value, i.e., $p = \frac{525}{2^{29}}$. We have also assumed different values of α and β., We start from β parameter, which expresses the proportion of the number of analysed pairs to the number of all generated pairs, so it is an efficiency of filtration. Whole filtration we can do in the 3R attack is testing for all five s-boxes and check whether input difference obtained from ciphertexts and the characteristic may cause obtained output difference. If for these input and output differences in appropriate XOR profile it is zero entry then it means this input-output transition is impossible. If it happens even for one s-box out of five we know for sure that this pair is wrong and we discard it. Hence parameter β is a probability that for all five s-boxes simultaneously the input-output differences are possible. In average percentage of non-zero values in XOR profiles of all s-boxes of DES is 0.8. For five s-boxes we may assume $\beta = 0.8^5 \approx 0.32768$, but it is not sufficiently precise for us. We have to look closer at XOR profiles of s-boxes. In [3] percentages of non-zero values are given for all 8 s-boxes. We have determined these percentages with bigger precision. All results are given in Table 1.

Table 1
Percentage of non-zero values in XOR profiles of s-boxes

s-box	Percentage [3]	Precise values
$S1$	79.4	0.794921875
$S2^*$	78.6	0.786132813
$S3$	79.6	0.796875000
$S4$	68.5	0.685546875
$S5^*$	76.5	0.765625000
$S6^*$	80.4	0.804687500
$S7^*$	77.2	0.772460938
$S8^*$	77.1	0.771484375

In our attack we deal only with 5 s-boxes (denoted *) so parameter β is a product of corresponding values from the table and it equals:

$$\beta = 0.288631 < 0.8^5.$$

Than we calculate value of α. Parameter α means the number of subkeys suggested by one analysed pair. It is the average number of different subkeys found by key recovery algorithm for one non-discarded pair. To obtain α we will again use XOR profiles. For individual s-box the average number of found subkeys is equal to average entry in the XOR profile of the s-box. It is equal 64 (the sum of entries in every row) divided by 16 (the number of columns) hence it is 4. But after filtration we do not take impossible transitions in to account. So we should calculate the average only from non-zero entries. The values of α calculated in that way are given in Table 2. As we can see they are different for different s-boxes.

Table 2
The average of non-zero entries in XOR profile of s-boxes

s-box	Average of non-zero values
$S1$	5.031941032
$S2^*$	5.088198758
$S3$	5.019607843
$S4$	5.834757835
$S5^*$	5.224489796
$S6^*$	4.970873786
$S7^*$	5.178255373
$S8^*$	5.184810127

Similarly like for β, α is a product of values for five s-boxes denoted by * and it equals:

$$\alpha = 3547.782689.$$

Product of α and β is equal to $\alpha\beta = 1024 = 4^5$, so previous estimation was also very precise. But the second method is more universal; because it calculates values of α and β separately, however it is also needs more work.
Now we know the values of all parameters so we can calculate S/N again:

$$S/N = \frac{2^{30} \cdot \frac{525}{2^{29}}}{0.288631 \cdot 3547.782689} = \frac{1050}{1024} = 1.025390625.$$

This value is different from previous on the third significant position although we actually have changed only the probability p.
Very interesting thing is that experiments (cf. Section 6) provide different value of β parameter. We have analysed different number of pairs and obtain the number of pairs non-discarded in filtration process. Proportion of these numbers determines β.
From Table 3 we can see that value of the parameter is rather constant. Small differences are consequences of probabilistic behaviour of differential cryptanalysis. It is also slightly different from theoretically obtained value. It may be explained as follows. We generate pairs of plaintexts not actually in random. First plaintext from the pair is generated randomly but second one depends on the characteristic. Than pairs of ciphertexts are not random and differential cryptanalysis can work at all. Theoretical estimation of β was obtained for fully random values. And here this difference appears. About $1/p$ pairs are good and survive filtration for sure, some portion of pairs may follow

the characteristic for a few rounds and have more chances to survive filtration then fully random ciphertexts. Hence in practice more pairs remain after filtration and parameter β is bigger then theoretical estimation made for fully random pairs.

Table 3
Parameter β obtained in experiments

No.	β
1	0.2916292
2	0.2915709
3	0.2915562
4	0.2914597
5	0.2920615
6	0.2915714

Parameter α obtained in those experiments was accurate with its theoretical value.
We put value of β provided by experiments to the formula of the S/N and we have:

$$S/N = \frac{2^{30} \cdot \frac{525}{2^{29}}}{0.2915 \cdot 3547.782689} = 1.015298465\,.$$

Value of the S/N obtained in this way we will use in further considerations.

4.4. The method of determination the number of pairs needed to the attack based on the signal to noise ratio

We have determined as precise as possible the S/N ratio. Now we will use the ratio to estimate the number of required pairs. If the S/N is much bigger then 1 we can assume that even 3–4 good pairs are sufficient to successful attack. But when the S/N is close to 1 like in that case the number of required good pairs is much bigger. In-advance assumption that 20, 30 or 40 good pairs are sufficient must be verified in certain attack. The attack on 9 rounds of DES need more then suggested in [3] 30 or 40 good pairs to uniquely identify the correct key. Sometimes these numbers are sufficient but it happens too rarely. Rough estimation of efficiency of attack with these numbers of pairs is smaller then 50%. So how many pairs are needed to significantly increase this efficiency?
We want to set the number of required pairs to be as small as possible but to be sufficient to uniquely identify the correct key. To recall, found key is a value, which has occurred the most frequently. It means that we search the counter of occurrences of all keys for the maximum value. If a noise is high the maximum value may not correspond to the correct key but some other key. The number of occurrences of the correct key may be second or third or next value in the counter. In that case attack ends with failure. In our attack the S/N ratio is greater than 1 (and it is possible at all) so increasing number of generated and analysed pairs tends to increased number of occurrences of correct key. That number increases faster than average number of occurrences of incorrect key (noise). So the number of pairs to analyse should be sufficiently big to assure that number of occurrences of the correct key will be maximum value in the counter. It means that number will be greater than number of occurrences of all other keys with big probability. Due to the probabilistic nature of our problem values in the counter are different in different experiments. And we can use only average numbers, which are easy to determine. We cannot predict the maximum value of noise or signal in the counter but we know in average how many good pairs were analysed. Also we know that every good pair gives one good key. We know the average value in the counter (level of a noise) as well. Now we want to be sure that number of occurrences of correct key would "stand out of noise". It means it would be greater even by 1 than other values in the counter. Hence we have to set the number of required pairs in order to the expected number of good pairs and occurrences of good key as well be greater than the expected average number in the counter. The expected number of good pairs can be calculated as a product of the number of all analysed pairs and the probability of the characteristic: $m \cdot p$. It is the numerator of the S/N ratio. The expected average number in the counter can be calculated as: $(m \cdot \alpha \cdot \beta)/2^k$ (the denominator of the S/N). Now we find such m that the numerator $(m \cdot p)$ is greater at least by 1 than the denominator. We start with $m = 30\,000\,000$ because we know that value is too small. We assume step 5 million and check successive values of m. Results of our searching are given in Table 4.
The first row gives the number of pairs used in the attack. The second counts the average number in the counter, third – number of good pairs. The fourth row is a proportion of above rows (row 2/row 3), so it is the S/N ratio actually. The last row expresses the difference: row 3 – row 2. As we can see the difference is greater than 1 for $m = 70$ million and we end our search with that value. We believe that in some sense the last row expresses the success of attack with corresponding number of pairs.
Precise value of m can be also calculated by using the S/N ratio. The number of occurrence of good key grows faster than the average number in the counter by factor equals to the S/N ratio. We want the signal be greater by one than noise. We put the nominator to be by 1 greater than the denominator, hence:

$$mp - \frac{mp}{S/N} > 1 \Rightarrow \frac{(S/N-1)mp}{S/N} > 1 \Rightarrow m > \frac{S/N}{S/N-1}\,\frac{1}{p}\,.$$

In that way we obtain the estimation on the number of required pairs to successful attack:

$$m > \frac{S/N}{(S/N-1)p}\,.$$

As we can see, with the growth of the S/N ratio the number m tends to $1/p$.

Table 4
Results of searching for required number of pairs

1.	Number of pairs m	30000000	35000000	40000000	45000000	50000000
2.	Average in counter	28.89462	33.71038743	38.52616	43.34192669	48.15769633
3.	Number good pairs	29.33666	34.22610462	39.11555	44.00499165	48.89443517
4.	The S/N ratio	1.015298	1.015298465	1.015298	1.015298465	1.015298465
5.	Difference	0.44204	0.51571719	0.58939	0.66306496	0.73673884
1.	Number of pairs m	55000000	60000000	65000000	67866654	70000000
2.	Average in counter	52.97346596	57.7892356	62.60500522	65.3660333	67.42077
3.	Number good pairs	53.78387868	58.6733222	63.56276572	66.3660333	68.45221
4.	The S/N ratio	1.015298465	1.015298465	1.015298465	1.015298465	1.015298
5.	Difference	0.81041273	0.884087	0.95776049	1	1.03143

The estimation can be useful in every attack, especially when the S/N ratio is close but greater than 1. Efficacy of attack with that number of pairs is very high (cf. Section 6) but still less than 100%. Due to the probabilistic nature of the problem an experiment in which that number is too small always may happen. But now it will happen rarely. In that case we of course can use more pairs.
In our attack we have:

$$m = \frac{1.015298465 \cdot 2^{29}}{(1.015298465 - 1) \cdot 525} = 67866653.95\,.$$

Hence we need above 67 million pairs. It confirms the results given in the Table 4.
Using the above estimation we can make quite interesting observation. Namely, if we use the estimation with very first calculated value of S/N ratio [3] we will obtain $m > 22074391.17$. It means that for this value of the S/N ratio 30 million pairs would be sufficient.

5. Implementation issues

Implementation and performance of considered attack is possible now thanks to progress in computational power of computers (processor speed, capacity of operational and disc memory). But still there are some problems and restrictions we have to solve. Main problem is a size of memory to store the counter.
We count the number of occurrences of all 30-bit subkeys, so we need the counter, which consists 2^{30} at least 8-bit values. It requires operational memory of size 1 GB. In the attack on 8 rounds of DES the problem was solved by dividing it on two less complicated subproblems [2]. In the case of 9 rounds this solution is impossible. Computer with 1 GB of operational memory is still unavailable due to high costs. The problem was solved on computer with operational memory of size 256 MB by time-memory trade off 256 MB of memory allows on using a counter with 2^{28} elements. The space of 2^{30} was divided on four separate subspaces of size 2^{28} depending on two most significant bits of 30-bit key. Whole process of counting keys is divided on four substages.
In first substage we generate required number of pairs. Than we perform key recovery algorithm and save in temporary file only non-discarded pairs (ciphertexts actually). It requires $2 \cdot \beta \cdot m \cdot 8$ bytes of disc memory and for $m = 70$ million it makes about 324 MB and now it is easily available. In that smaller counter we put only keys with the same two significant bits equal for example 00. In the counter we search for the maximum. If the distinct maximum exists we assume it corresponds with the correct key and we end this stage. The distinct maximum means that it is only one maximum value and the value is significantly greater than any others. In the other case we proceed to next substage. We put into the new counter only keys with the same two significant bits equal now for example 01.
In the second substage we use saved in the first substage non-discarded pairs. We do not perform the filtration again what significantly reducing the time of the substage. Than we again search for the distinct maximum in the new counter taking into account maximum value from the first counter. If the distinct maximum exists we finish with that value of key. In the other case we proceed to third and forth if it is needed. The time of that stage may differ significantly with respect to the number of substages performed until the key was found. In average we perform two substages. That approach makes the issues of required number of pairs very complicated. If the number appeared too small increasing it is very troublesome and we must repeat whole attack actually. As we can see it is very important to set the appropriate number of required pairs at the beginning. From the other hand, the number should be as small as possible to decrease the time of attack.
The second and very unexpected problem, which showed out, was generating the pairs of plaintexts. As we stated we need 70 million pairs, so we have to generate above $2^{26} \cdot 2^3 = 2^{29}$ bytes in random. The period of standard pseudorandom number generator in used programming language (C++) should be 2^{31}, but for the least significant byte it is smaller and it is only 2^{27} (it is a discovered error

Table 5

First results

Number of generated pairs [million]	Number of non-discarded pairs	β	Max in counter	Number of occurrences of correct key	Result
30	8743769	0.2914580	64 (4 times)	63	Failure
35	10202090	0.2914883	77	77	Success
35	10203548	0.2915299	73	59	Failure
35	10205025	0.2915721	81	81	Success
35	10204767	0.2915648	71	61	Failure
35	10203362	0.2915246	70 (3 times)	70	Failure
35	10200564	0.2914447	76	76	Success
35	10204390	0.2915537	73 (2 times)	64	Failure
35	10207181	0.2916337	72 (2 times)	59	Failure
35	10205876	0.2915965	75	48	Failure
35	10201090	0.2914597	73	60	Failure
40	11655712	0.2913928	88	88	Success
40	11658656	0.2914664	92	92	Success
40	11662246	0.2915562	87	87	Success
40	11661319	0.2915330	82	76	Failure
40	11665598	0.2916400	79 (2 times)	64	Failure
40	11662246	0.2915562	84	84	Success
50	14576501	0.2914666	99	99	Success
50	14573329	0.2916178	105	105	Success
50	14580889	0.2916178	98	80	Failure
50	14575285	0.2915057	95	88	Failure
50	14586415	0.2917283	109	109	Success
50	14576447	0.2915289	101	101	Success
50	14579432	0.2915886	94	85	Failure
50	14579003	0.2915801	100	100	Success
50	14575915	0.2915183	95	94	Failure

of compiler!). So we have to use our own pseudorandom number generator with the suitable period, which generates numbers with uniform distribution. We did not need any cryptographically strong generator but only fast one so we used ordinary linear feedback shift register (LFSR) with length 64 bits, what made its period equals to $2^{64} - 1$ bits. That problem is irrelevant from cryptanalytic point of view but we would like to point out that in the case of such huge amounts of data similar problems may completely warp the results of the cryptanalysis.

6. Results

Now we present the results of performance of considered attack. The attack was implemented in C++ language in Borland C++ Builder 5 on a computer with processor Celeron II 1.3 GHz and 320 MB of operational memory. We used implementation of 9 rounds of DES running with speed 3.2 million blocks per second what makes throughoutput 200 Mbit/s.

We start with attacks with too small number of pairs. Table 5 shows the results of attacks with 30–50 million pairs. The first column presents the number of generated pairs. The second expresses the number of pairs, which survive the filtration. The third is a proportion of two previous columns and it is an efficacy of filtration (parameter β). In the next column the maximum value in the counter is given. The number in the parenthesis means how many times the maximum appeared if more than one. The fifth column gives the number of occurrences of the correct key. In the last column the result of the attack (success or failure) is given. The result can be derived from two previous columns.

From the table we can roughly estimate the probability of success of the attack. The probability of success of attack with 30 or 35 million pairs is smaller than 1/3, with 40 or 50 million pairs slightly exceeds 1/2.

Table 6 presents in similar way the results of attacks with 70 million pairs. Additionally the time of performance of substages is given.

Table 6

Main results

Number of all pairs [million]	Number of non-discarded pairs	β	Time of substages [s]				Time of II stage [s]	Time of attack [s]	Max	Number of occurrences of correct key	Result
			1	2	3	4					
70	20411466	0.2915924	7391	4987	4977	–	138	17493	139	139	Success
70	20407887	0.2915412	7395	4979	4983	–	6	17363	144	144	Success
70	20400900	0.2914414	7374	–	–	–	245	7619	127	127	Success
70	20410516	0.2915788	7384	–	–	–	84	7468	134	134	Success
70	20408476	0.2915497	7363	–	–	–	29	7392	128	128	Success
70	20408331	0.2915476	7382	–	–	–	85	7467	145	145	Success
70	20402070	0.2914581	7358	4982	4948	6085	313	23686	121	116	Failure
70	20414410	0.2916344	7376	–	–	–	273	7649	149	149	Success
70	20411673	0.2915953	7384	4945	4955	4951	136	22371	143	143	Success
70	20404332	0.2914905	7360	4935	–	–	72	12367	133	133	Success
70	20405427	0.2915061	7353	4938	–	–	211	12502	132	132	Success
70	20414960	0.2916423	7389	4944	–	–	155	12488	123	123	Success
70	20404748	0.2914964	7391	–	–	–	85	7476	133	133	Success
70	20409296	0.2915614	7353	4932	4927	–	89	17301	135	135	Success
70	20415691	0.2916527	7365	4916	–	–	196	12477	148	148	Success
70	20408939	0.2915563	7455	–	–	–	17	7472	126	126	Success
70	20403624	0.2914803	6792	–	–	–	210	7002	151	151	Success
70	20409608	0.2915658	6804	4729	4707	4710	152	21102	119	118	Failure
70	20404448	0.2914921	6766	–	–	–	232	6998	121	–	Failure
70	20412821	0.2916117	6768	–	–	–	182	6950	144	144	Success
70	20413305	0.2916186	6764	–	–	–	77	6841	152	152	Success
70	20404312	0.2914902	6772	–	–	–	13	6785	149	149	Success
70	20413398	0.2916200	6769	4709	4696	–	28	16202	146	146	Success
Average		**0.2915574**	**7167**	**4892**	**4847**	**5249**	**132**	**11500**	**Succ./fail.**		**20/3**

Lack of given time of any substage means that substage was not necessary because the maximum was found in previous substage. The first substage is always about 1.5 times longer than others. It results from applied procedure of counting keys. In the first stage we generate all pairs, than we perform the filtration and save them. In next substages we only analyse non-discarded pairs what is significantly faster.

As we can see only 3 out of 23 attacks have ended with failure. In the two first cases all four substages were performed, but in any of them distinct maximum was not found, and the correct key occurred more rare than others. In the third case distinct maximum was found but it did not correspond to the correct key. The efficiency of the attack with 70 million pairs we can consider as very high and close to 90% even for a few dozen experiments.

After assuming that 70 million is the proper number of required pairs we can perform a full attack. It means after finding the 30 bits of main key in the first stage we can find remaining 26 bits by exhaustive search of space 2^{26}. It was now a simple and fast task. The column "Time of II stage" in Table 6 gives time (in seconds) of exhaustive search that was needed to find remaining 26 bits of main key. As we can see that search takes not longer than 5 minutes. The column "Time of attack" gives a total time of the attack calculated as the sum of fives values in the previous columns.

In the cases of first two failures the correct key occurred too less times to be found. So all four substages were performed and entire search in the II stage (313 and 152 seconds), but without success. Of course if first 30 bits are wrong we can never adjust the last 26 bits to get the correct key.

The last failure was of different type. In the initial substages (in the first in the certain case) a distinct maximum was found, but it did not correspond to the correct key. Algorithm ended without performing the next substages in which the correct key should be found. That way we did not found the number of occurrences of the correct key. That failure is not based on too less number of analysed pairs, but it comes from extorted dividing the first stage on four

Table 7
Final results

Number of all pairs [million]	Number of non-discarded pairs	β	Time of substages [s]				Time of II stage [s]	Time of attack [s]	Max	Number of occurrences of correct key	Result
			1	2	3	4					
70	20407072	0.2915296	6799	4772	4782	4793	151	21297	124	124	Failure
70	20411849	0.2915978	6847	–	–	–	31	6878	130	130	Success
70	20412699	0.2916100	6837	–	–	–	138	6975	133	133	Success
70	20414234	0.2916319	6842	4934	4926	4940	73	21715	145	145	Success
70	20407731	0.2915390	6838	4907	4932	4928	179	21784	131	141	Success
70	20402279	0.2914611	6834	4919	4933	4917	152	21755	124	117	Failure
70	20409520	0.2915646	6841	–	–	–	233	7074	125	–	Failure
70	20407149	0.2915317	6829	–	–	–	69	6898	156	156	Success
70	20409663	0.2915666	6835	–	–	–	152	6987	156	156	Success
70	20413565	0.2916224	7030	4776	4896	4812	151	21665	121	115	Failure
70	20412273	0.2916049	6827	4797	4805	–	142	16571	128	128	Success
70	20412370	0.2916053	6867	5002	5014	5015	212	22110	122	122	Failure
70	20405958	0.2915137	8277	5406	5118	4886	41	23728	130	130	Success
70	20412912	0.2916130	4704	–	–	–	225	4929	139	139	Success
70	20416821	0.2916689	6740	4699	4695	–	163	16297	144	144	Success
70	20406430	0.2915204	6818	4832	4855	4942	152	21599	123	120	Failure
70	20401766	0.2914548	6827	4809	–	–	226	11862	149	149	Success
70	20409267	0.2915610	6855	–	–	–	71	6926	136	136	Success
70	20409910	0.2915701	6832	4808	4837	4819	69	21365	148	148	Success
70	20411009	0.2915858	6830	–	–	–	205	7035	149	149	Success
70	20406152	0.2915165	6880	–	–	–	109	6989	138	138	Success
70	20407093	0.2915299	7576	5120	5120	4882	68	22766	144	144	Success
70	20399601	0.2914229	6712	4806	4804	4817	146	21285	130	130	Success
70	20410754	0.2915822	6838	–	–	–	255	7093	136	136	Success
70	20406290	0.2915184	7796	4959	5184	–	90	18029	133	133	Success
70	20407788	0.29153983	7571	5119	5119	4895	197	22901	152	152	Success
70	20416450	0.29166357	6722	–	–	–	222	6944	142	142	Success
70	20413332	0.29161903	6840	4937	4929	4943	150	21799	146	146	Success
70	20408688	0.29155269	8576	5337	5115	–	208	19236	141	141	Success
70	20408957	0.29155653	6835	4925	4931	–	244	16935	136	136	Success
70	20408126	0.29154466	6831	4925	4933	4922	220	21831	135	135	Success
70	20407964	0.29154234	6836	4930	4946	4948	46	21706	154	154	Success
70	20408866	0.29155523	6835	4941	–	–	31	11807	128	128	Success
Average		**0.29156041**	**6938**	**4939**	**4944**	**4897**	**146**	**15599**	**Succ./fail.**		**27/6**

substages. In that case we should despite to find suspect key in the initial substages continue with next substages. So the open problem appears: should we always perform all four substages what will increasing time of the attack significantly or like it was done stop after finding the first distinct maximum what is faster but generate above failures?

The way of omitting that problem is fixing a threshold on the maximum value in the counter. If the found maximum is lower than the threshold we will continue with the next substages. As the performed experiments show "the level of noise" which is the maximum number of occurrences of incorrect key does not exceed 125. However the correct key usually occurs (if we reached adequate substage) more often, and even more than 130 times. Hence we can assume that if we find the distinct maximum but lower than 125 that value does not correspond to the correct key and we will continue with next substages. The results of attacks performed with that rule are presented in Table 7.

Introducing the threshold almost eliminated the failures, which comes from dividing problem on substages. Failures, which appeared now, derive from too small number of occurrences of the correct key. In those cases we should generate more pairs to analyse. The exception is the first failure and the failure where the number of occurrences of the correct key is not given. In the first case the maximum value in the counter corresponded to the correct key but it did not exceeded the fixed threshold and was not taken in to account. In that case even without applying the "threshold" rule attack would end with failure, because found maximum was not distinct, the second biggest value was only smaller by 1. In the second case the distinct maximum was found and it exceeded the threshold but it was not the correct key. In that case the threshold should be higher. But the higher threshold may cause more failures of the first (previous) type. Fixing the threshold is very hard and important case. Increasing the threshold will reduce the number of failures of second type but will increase the number of failures of the first type and vice versa. However the small number of total failures in our experiments let us consider that we fixed the threshold correctly.

The efficiency of the attack we can roughly approximate on 80%, 6 failures in 33 tries, and the average time of performance of entire attack was 15 600 seconds, which is about 4 hours and 20 minutes. It is very short time for recovering full 56-bit key of 9-round algorithm.

Acknowledgement

This work has been partly supported by Polish Committee of Science Research project number 0 T00A 020 25 and partly supported by the European Commission under contract IST 2002-507932 (ECRYPT).

References

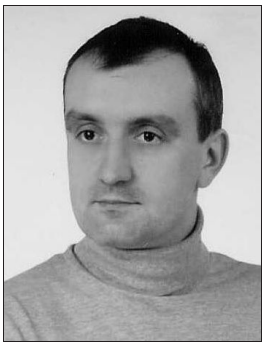
[1] E. Biham, V. Furman, M. Misztal, and V. Rijmen, "Differential cryptanalysis of Q", in *Fast Software Encryption: 8th International Workshop, FSE 2001, Yokohama, Japan, April 2–4, 2001*, M. Matsui, Ed., *Lecture Notes in Computer Science*. Berlin [etc.]: Springer-Verlag, 2002, vol. 2355, pp. 174–186.

[2] M. Misztal, "Praktyczna kryptoanaliza różnicowa algorytmu DES zredukowanego do 8 rund", *Bull. WAT Cryptology Part I*, vol. XLVII, no. 10(566), pp. 125–146, 1999 (in Polish).

[3] E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*. New York: Springer-Verlag, 1993.

[4] D. Kwiatkowski, "Implementacja i kryptoanaliza wybranych szyfrów blokowych", Warszawa, Wojskowa Akademia Techniczna, Wydział Cybernetyki, 1998, Master thesis (in Polish).

[5] B. Schneier, *Kryptografia dla praktyków. Protokoły, algorytmy i programy źródłowe w języku C*. Warszawa: WNT, 2002 (in Polish).



Michał Misztal was born in 1973 in Kielce, Poland. He got his M.Sc. in 1997 from Faculty of Cybernetics of Military University of Technology (MUT), Warsaw. He has studied on "cryptology" specialty in the individual course. He works as an Assistant in the Institute of Mathematics and Cryptology on Faculty of Cybernetics MUT. He conducts tutorials and lectures on mathematics and linear algebra, but also on profiled by the Institute "cryptology" specialty on such subjects like cryptanalysis of block and stream ciphers, differential and linear cryptanalysis and designing of block ciphers. He is the co-author of handbook entitled "Introduction to Cryptology" and the author of several papers published among others in the bulletin of MUT. He has also given many lectures on scientific conferences devoted to cryptology.

e-mail: mmisztal@wat.edu.pl
Institute of Mathematics and Cryptology
Faculty of Cybernetics
Military University of Technology
S. Kaliskiego st 2
00-908 Warsaw, Poland

Information for authors

Journal of Telecommunications and Information Technology **(JTIT)** is published quarterly. It comprises original contributions, both regular papers and letters, dealing with a wide range of topics related to telecommunications and information technology. **All regular papers and letters are subject to peer review**. Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

The JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

Suitable research-related papers should hold the potential to advance the technological base of telecommunications and information technology. Tutorial and review papers are published only by invitation.

Manuscript. Papers published by invitation and regular papers should contain up to 15 and 8 printed pages respectively (one printed page corresponds approximately to 3 double-spaced pages of manuscript where one page contains approximately 2000 characters). An original and one copy of the manuscript should be submitted, each completed with all illustrations, tables and figure captions attached at the end of the paper. Tables and figures should be numbered consecutively with Arabic numerals. The manuscript should include an abstract about 100 words long and the relevant keywords. The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results.
The manuscript should be double-spaced on one side of each A4 sheet (210 × 297 mm) only. Use of computer notation such as Fortran, Matlab, Mathematica, etc., for formulae, indices and the like is not acceptable and will result in automatic rejection of the manuscript.

Illustrations. Original illustrations should be submitted. Drawings in Corel Draw and Postscript formats are preferred. Colour illustrations are accepted only under exceptional circumstances. Lettering should be large enough to be readily legible when drawing is reduced to two- or one-column width, which often means shrinking to 1/4th of original size. Photographs should be used sparingly. All photographs should be delivered in electronic formats (TIFF, JPG, PNG or BMP). Page numbers including tables and illustrations (which should be grouped at the end) should be put in a single series, with no numbers skipped.

References. All references should be marked in the text by Arabic numerals in square brackets and listed at the end of the paper in order of their appearance in the text, including exclusively publications cited inside. Samples of correct formats for various types of references are presented below:

[1] Y. Namihira, "Relationship between nonlinear effective area and mode field diameter for dispersion shifted fibres", *Electron. Lett.*, vol. 30, no. 3, pp. 262–264, 1994.

[2] C. Kittel, *Introduction to Solid State Physics*. New York: Wiley, 1986.

[3] S. Demri and E. Orłowska,"Informational representability: Abstract models versus concrete models", in *Fuzzy Sets, Logics and Knowledge-Based Reasoning*, D. Dubois and H. Prade, Eds. Dordrecht: Kluwer, 1999, pp. 301–314.

Biographies and photographs of authors. A brief professional author's biography of up to 100 words and a photo of each author should be included with the manuscript. A printed photo, minimum size 35 × 45 mm, is acceptable. The photo may be supplied as image file.

Electronic form. TEX and LATEX are preferable, standard Microsoft Word format (.doc) is acceptable. The JTIT LATEX style file is available to authors. The file(s) should be submitted by e-mail or on a floppy disk or CD together with the hard copy of the manuscript. It is important to ensure that the diskette version and the printed version are identical. The diskette should be labelled with the following information: a) the operating system and word processing software used; b) in case of UNIX media, the method of extraction (i.e., tar) applied; c) file name(s) related to manuscript. The diskette or CD should be properly packed in order to avoid possible damage in the mail.

Galley proofs. Authors should return proofs as soon as possible. In other cases, the article will be proof-read against manuscript by the editor and printed without the author's corrections. Remarks to the errata should be provided within two weeks after receiving the offprint.

Copyright. Manuscript submitted to JTIT should not be published or simultaneously submitted for publication elsewhere. By submitting a manuscript, the author(s) agree to automatically transfer the copyright for their article to the publisher, if and when the article is accepted for publication. The copyright comprises the exclusive rights to reproduce and distribute the article, including reprints and all translation rights. No part of the present JTIT should not be reproduced in any form nor transmitted or translated into a machine language without prior written consent of the publisher.

A copy of the JTIT is provided to each author of paper published.
